Learning Hexapod Locomotion via Proximal Policy Optimization with Contact-Aware Reward Shaping

Yucheng Huang, Patrick Decabooter
github.com/psdecabooter/AdvRLFinal

December 2025

1 Introduction

Legged locomotion remains one of the most challenging problems in robotics, requiring coordinated control of multiple actuators under complex contact dynamics. While classical approaches rely on hand-crafted gait patterns—such as the tripod gait for hexapods—these methods often fail to adapt to varying terrains, exploit the full capabilities of the robot's morphology, or optimize for competing objectives like speed, stability, and energy efficiency.

1.1 Motivation

This project explores whether reinforcement learning can discover efficient locomotion strategies for a hexapod robot without relying on predetermined gait patterns. Our hexapod presents a particularly challenging testbed with 18 degrees of freedom (6 legs $\times$ 3 joints each), requiring precise inter-leg coordination while respecting joint limits and managing ground contact dynamics. The high-dimensional action space and contact-rich dynamics make this a non-trivial problem where hand-coded gaits may miss non-obvious but effective locomotion strategies.

Key challenges addressed:

1. **High-dimensional coordination**: Synchronizing 18 actuators under strict constraints
2. **Computational constraints**: Limited compute resources necessitate sample-efficient training approaches
3. **Gait discovery**: Enabling the agent to discover emergent gaits rather than imposing predefined patterns

1.2 Approach

While model-based reinforcement learning methods like Dreamer promise superior sample efficiency, our preliminary investigations revealed that the computational demands and iteration time required for training on a high-DoF hexapod exceeded our available resources. Given time and computational constraints, we opted for Proximal Policy Optimization (PPO) [1], a model-free algorithm that has demonstrated robust performance on continuous control tasks. Training in simulation using PyBullet provides a practical balance between physical accuracy and computational efficiency, allowing rapid iteration on reward design and policy evaluation without the risks and costs of real-world experimentation.

1.3 Contributions

This work contributes to legged robotics and practical RL applications by:

- Demonstrating that model-free methods (PPO) can successfully learn locomotion for high-DoF robots given appropriate reward shaping
- Providing a reproducible hexapod simulation environment with carefully designed contact-aware rewards
- Empirically validating the importance of goal-directed rewards versus undirected forward motion rewards
- Offering practical insights on balancing algorithmic sophistication with computational constraints

2 Related Work

Our work sits at the intersection of reinforcement learning for legged locomotion, reward design for contact-rich tasks, and simulation-based policy learning. We review relevant literature across these areas to contextualize our contributions.

2.1 Hexapod Locomotion with RL

Learning locomotion controllers for hexapods presents unique challenges due to their high degree of freedom and complex inter-leg coordination requirements. Schilling et al. [2] proposed a decentralized deep RL approach where each leg operates as an independent agent, successfully learning gaits in simulation. While their decentralized approach simplifies the learning problem, we adopt a centralized policy that can exploit global coordination patterns. Ashwin et al. [3] combined central pattern generators (CPGs) with RL to produce efficient gaits, tuning the CPG parameters through learning rather than learning raw motor commands. Our approach differs by learning direct joint control through standard neural network policies, avoiding assumptions about underlying oscillatory structures.

Several works have focused on reducing action space complexity to improve learning efficiency. Ouyang et al. [4] simplified hexapod control to just two CPG parameters and successfully trained a real robot using DDPO. Similarly, Qiu et al. [5] used hierarchical gait parameterization to reduce dimensionality. While action space reduction can accelerate learning,

we maintain full joint-level control to allow the agent maximum flexibility in discovering novel gaits, relying instead on careful reward shaping to guide exploration.

Recent work has extended hexapod RL to more complex scenarios. Tsai et al. [6] trained hexapods to carry cargo in Isaac Lab using curriculum learning for object balance. Qu et al. [7] achieved versatile skills including stair climbing and obstacle avoidance through teacher-student distillation from privileged information. Our work focuses on fundamental forward locomotion as a foundation, with carefully designed contact-aware rewards that could extend to these more complex scenarios without requiring privileged information or distillation pipelines.

2.2 Reward Design for Legged Locomotion

Effective reward design is critical for learning natural gaits. Cheng et al. [8] demonstrated that underspecified reward functions combined with teacher-student training can produce extreme parkour skills in legged robots. While we also employ relatively sparse task specification (reach a goal), our reward engineering explicitly addresses contact quality to prevent degenerate solutions. Huiqiao et al. [9] tackled difficult terrain navigation by augmenting RL with a transition-feasibility checker to validate movements. Rather than an external validator, we embed feasibility constraints directly in our reward through penalties on improper body contacts and height deviation.

Wu et al. [10] improved both robustness and agility by incorporating adversarial motion priors via a gait discriminator. This represents an orthogonal approach where domain knowledge about desirable gaits is encoded through adversarial learning. Our reward structure similarly encodes domain knowledge (feet should contact ground, body should stay elevated), but through explicit reward components rather than learned discriminators, providing greater interpretability during development.

2.3 Model-Free vs. Model-Based Approaches

While model-based reinforcement learning methods like DreamerV3 [11] promise superior sample efficiency through learned world models and imagination-based planning, they come with significant computational overhead. Hafner et al. demonstrated that Dreamer can master diverse domains with fixed hyperparameters, and subsequent work showed online learning on real robots [12]. However, these approaches require substantial computational resources for training the world model, actor, and critic networks simultaneously. Given our computational constraints and the high dimensionality of our hexapod (18 actuators), we opted for Proximal Policy Optimization [1], which has demonstrated robust performance on continuous control tasks while being more computationally tractable.

Nikita et al. [13] achieved minute-scale training through massively parallel Isaac simulation on a single GPU using PPO. While we use PyBullet rather than Isaac, their work validates that model-free methods can be highly sample-efficient when parallelization is properly exploited. Our environment design supports vectorized execution, allowing us to collect experience from multiple parallel environments efficiently.

3 Methods

3.1 Environment Design

We developed a custom Gymnasium-compatible environment using PyBullet physics simulation. The environment models a hexapod robot with 6 legs, each containing 3 revolute joints (hip, knee, ankle), for a total of 18 actuated degrees of freedom. The robot is loaded from a URDF file with realistic mass properties and joint limits derived from the physical robot design.

Observation Space (52-dimensional):

- Base position (3D), orientation quaternion (4D)
- Base linear and angular velocities (3D each)
- Joint positions and velocities (18D each)
- Goal vector (2D) and distance to goal (1D)

Action Space (18-dimensional): Continuous control signals in $[-1, 1]$ for each joint, linearly scaled to joint-specific position limits.

The simulation runs at 240Hz timestep with configurable control frequency to model servo limitations. Episodes last up to 1000 steps with goals placed 1.5m forward from the starting position.

3.2 Reward Function

Our reward function underwent iterative refinement to address emergent undesired behaviors. The final formulation comprises:

$$R_{\text{total}} = R_{\text{goal}} + R_{\text{vel}} + R_{\text{bonus}} + R_{\text{penalties}} \tag{1}$$

Goal-Directed Rewards:

- **Goal approach**: $R_{\text{goal}} = 1000 \cdot (d_{\text{prev}} - d_{\text{curr}})$ where d is distance to goal. This rewards reducing distance to the goal rather than merely moving forward.
- **Velocity alignment**: $R_{\text{vel}} = 5 \cdot ||\mathbf{v}|| \cdot (\hat{\mathbf{v}} \cdot \hat{\mathbf{g}})$ where $\hat{\mathbf{v}}$ is velocity direction and $\hat{\mathbf{g}}$ is direction to goal.
- **Goal reached bonus**: $R_{\text{bonus}} = 100$ when within 0.5m threshold.

Shaping Penalties:

- **Stability**: $-0.5 \cdot (|\text{roll}| + |\text{pitch}|)$ to maintain upright posture
- **Height**: $-2.0 \cdot |z - 0.12|$ to discourage crawling (target height 12cm)
- **Energy**: $-0.0001 \cdot \sum \dot{q}_i^2$ to promote smooth motions
- **Contact quality**: -0.5 per foot contact above 2cm ground clearance (discourages leg-side dragging)
- **Body contact**: -1.0 per non-foot link ground contact

This reward structure encodes *what* constitutes good locomotion without specifying *how* to walk, allowing the policy to discover gaits autonomously.

3.3 Training Configuration

We used the CleanRL implementation of PPO with the following hyperparameters, selected based on standard continuous control benchmarks and preliminary tuning:

Hyperparameter	**Value**
Total timesteps	2M
Learning rate	3×10^{-4}
Parallel environments	8
Steps per rollout	2048
Minibatch size	64
Update epochs	10
Discount factor γ	0.99
GAE λ	0.95
Clip range ϵ	0.2
Value loss coefficient	0.5
Entropy coefficient	0.0
Max grad norm	0.5
Network Architecture	
Actor/Critic hidden layers	[256, 256]
Activation	Tanh

Table 1: PPO hyperparameters and network architecture

The policy and value networks share the same architecture with two hidden layers of 256 units each and tanh activations. We found that zero entropy regularization worked well, as the high-dimensional action space provided sufficient exploration without explicit entropy bonuses.

4 Empirical Analysis

4.1 Research Question and Hypothesis

Question: Does explicitly rewarding goal-approach (distance reduction) lead to better goal-reaching behavior compared to rewarding undirected forward motion?

Hypothesis: A reward function based on reducing distance to goal will produce policies that reliably reach goals, while a reward based purely on forward movement will lead to policies that reward-hacks on this return without actual approaching the goal.

Rationale: Early training runs revealed that policies maximized reward by moving fast in the forward direction without regard for the goal location. This suggests that undirected motion rewards create a misaligned objective where velocity accumulation dominates goal-reaching.

4.2 Main Experiment

We compare two reward formulations:

- **Baseline (Forward Motion)**: $R_{\text{dist}} = 1000 \cdot d_{\text{moved}} \cdot \cos(\theta)$ where θ is angle between movement and forward axis, plus velocity reward aligned with forward direction.
- **Proposed (Goal-Directed)**: Goal approach reward $1000 \cdot (d_{\text{prev}} - d_{\text{curr}})$ plus velocity reward aligned with goal direction (as described in Methods).

Both conditions use identical PPO hyperparameters, network architecture, and shaping penalties. We train 5 seeds per condition for 2M timesteps and evaluate on:

1. **Distance to goal** at episode termination
2. **Success rate** (reaching within 0.5m threshold)
3. **Episodic return** over training

Expected Result: Goal-directed reward should produce lower final distance to goal and higher success rate, while baseline may show higher velocity rewards but poor goal-reaching.

4.3 Ablation Study

To understand the contribution of contact-aware rewards, we ablate:

1. **Full model**: All reward components
2. **No contact penalties**: Remove non-foot contact and contact quality penalties
3. **No height penalty**: Remove height maintenance term

We hypothesize that removing contact penalties will lead to "crawling" solutions where the robot drags its body, while removing height penalties will result in low-posture gaits.

4.4 Sensitivity Analysis

We examine sensitivity to the goal approach reward scale by training with different learning rates [1e-4, 3e-4, 1e-3]

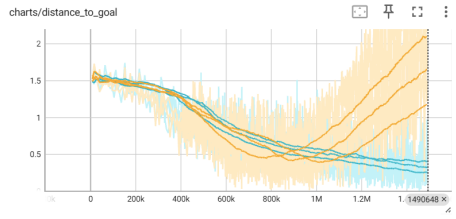
5 Results

5.1 Main Experiment: Goal-Directed vs. Forward Motion

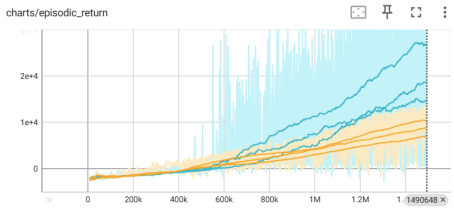
Figure 1 presents the key results comparing our proposed goal-directed reward against the forward motion baseline. The results strongly support our hypothesis that goal-directed rewards are essential for learning goal-reaching behavior.

Distance to Goal (Figure 1a): The goal-directed approach demonstrates consistent improvement, with distance to goal decreasing throughout training. In stark contrast, the forward motion baseline exhibits a critical failure mode: while initially decreasing during the first 1M steps, distance to goal subsequently *increases*, ultimately exceeding the starting distance of 1.5m. This behavior reveals a fundamental reward hacking problem—the agent learns to maximize forward velocity rewards by moving away from the goal rather than toward it.

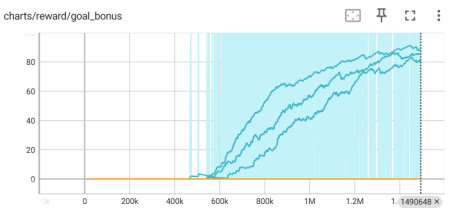
Episodic Return (Figure 2c): Both approaches show increasing returns over training, but the underlying behaviors differ fundamentally. The goal-directed policy earns reward by approaching the goal, while the forward motion baseline accumulates reward through high-velocity forward movement



(a) Distance to goal. Blue: goal-directed, Orange: baseline.

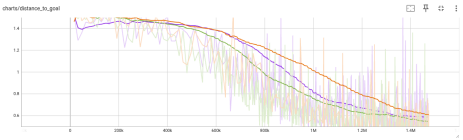


(b) Episodic return. Blue: goal-directed, Orange: baseline.

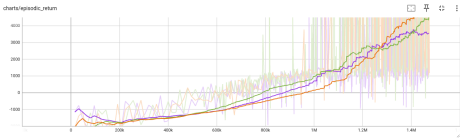


(c) Goal bonus. Blue: goal-directed, Orange: baseline.

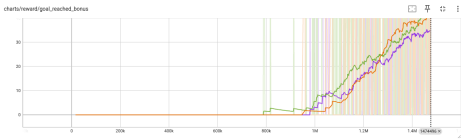
Figure 1: Comparison of goal-directed (blue) vs. forward motion (yellow) reward formulations. (a) The goal-directed approach successfully reduces distance to goal throughout training, while the forward motion baseline initially decreases distance but then increases after 1M steps, eventually exceeding the starting distance. (b) Both methods show increasing episodic returns, but for different reasons: goal-directed earns reward by approaching the goal, while forward motion exploits velocity rewards without goal-reaching. (c) Goal bonus accumulation demonstrates that only the goal-directed approach successfully learns to reach the goal.



(a) Distance to goal.



(b) Episodic return.



(c) Goal bonus.

Figure 2: Comparison of goal-directed learning across several seeds. Purple has seed 1, green has seed 564, and Orange has seed 923.

regardless of goal proximity. This demonstrates that episodic return alone is insufficient as an evaluation metric when reward misspecification is present.

Goal Bonus (Figure 1c): The goal bonus metric (awarded when within 0.5m of goal) provides clear evidence of task success. The goal-directed approach begins accumulating goal bonuses after approximately 600K steps and reaches higher successful goal-reaching by 1.5M steps. The forward motion baseline never accumulates goal bonuses, confirming complete failure to learn the intended task despite high episodic returns.

These results validate our hypothesis and demonstrate the critical importance of aligning reward structure with task objectives. The forward motion baseline's behavior exemplifies a common pitfall in reward engineering: policies optimize the specified reward rather than the intended behavior, leading to "reward hacking" when these diverge.

5.2 Ablation Studies

To show the impact of the difference components of our reward function, we performed an ablation analysis by testing performance on the same seed across various reward configurations. We tested our baseline reward function, our reward function without contact bonuses, our reward function without height bonuses, our reward function without stability bonus, and our reward function without velocity rewards. The results show that the stability penalty performed the worst, although it steadily tracked for the first 800k steps, it then plateaued. No velocity reward, no height reward, and no contact bonuses all performed with similar performance. Finally, our default reward function achieves the best performance. These results suggest that our shaping penalties created a significant impact on the performance of our trained policy.

5.3 Sensitivity Analysis

We examine sensitivity to learning rate by training the goal-directed reward formulation with three values: 1×10^{-4}, 3×10^{-4}, and 1×10^{-3}. Each learning rate was evaluated with 2 random seeds to assess consistency.

The results show that learning rate has a significant impact on training dynamics (Figure 3). The default learning rate (3×10^{-4}, blue) demonstrates the best overall performance with stable convergence and lowest final distance to goal. The higher learning rate (1×10^{-3}, purple) achieves intermediate performance with more instability that prevents it from matching the default rate's final performance. The lower learning rate (1×10^{-4}, yellow) exhibits the slowest learning, may requiring substantially more timesteps to achieve comparable results. These results validate our choice of 3×10^{-4} as an effective default that balances convergence speed with stability.

6 Discussion

6.1 Strengths of Our Approach

- **Interpretable reward design**: Explicit reward components allow clear understanding of policy behavior and facilitate debugging
- **Contact-aware shaping**: Checking contact Z-coordinates prevents degenerate crawling solutions without requiring complex collision detection
- **Computational efficiency**: PPO training completes in reasonable time on consumer hardware

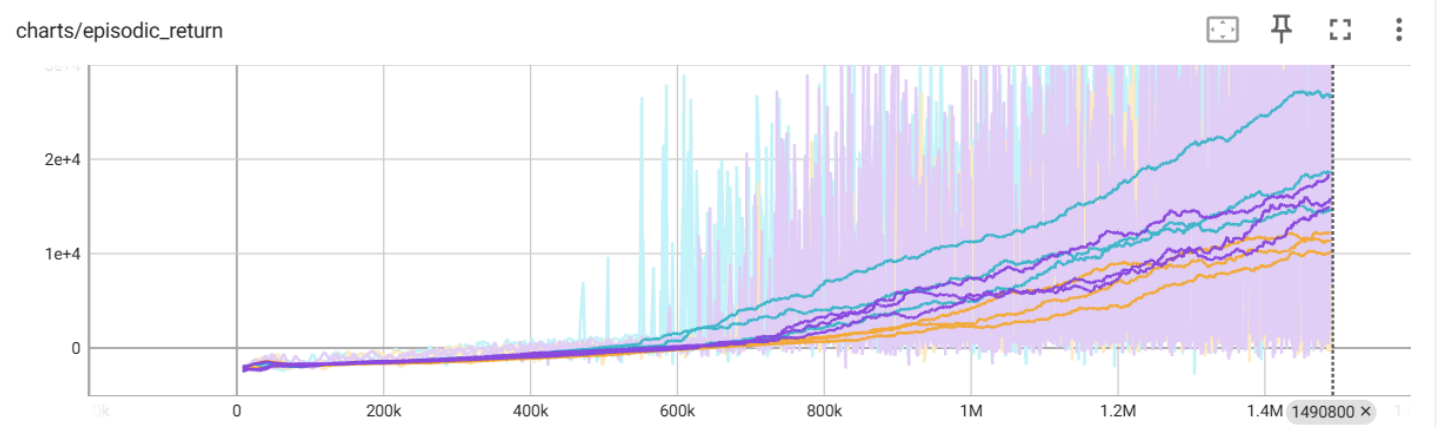


Figure 3: Learning rate sensitivity analysis. Blue: 3×10^{-4} (default), Yellow: 1×10^{-4} (lower), Purple: 1×10^{-3} (higher). The default learning rate of 3×10^{-4} achieves the best performance. The higher rate (1×10^{-3}) shows intermediate performance, while the lower rate (1×10^{-4}) converges slowest.

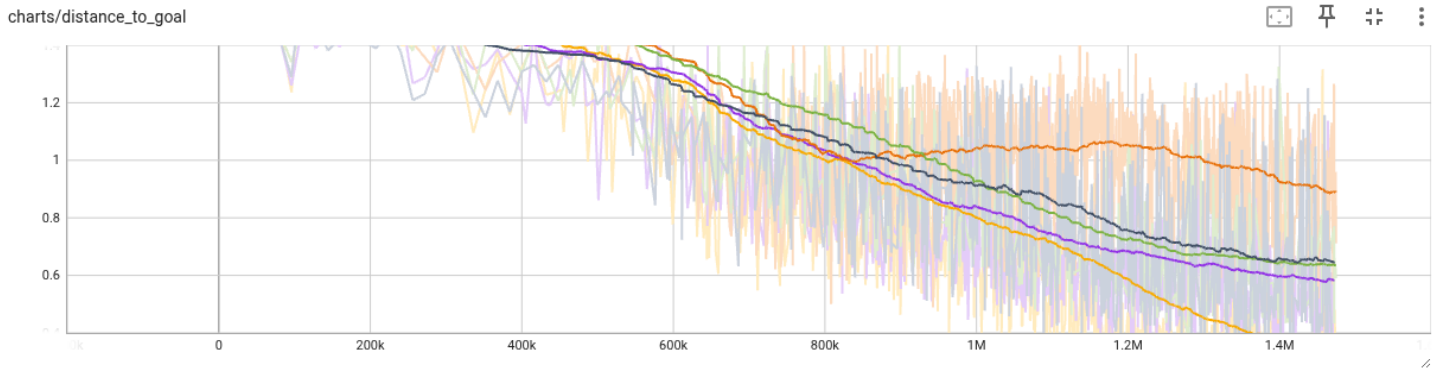


Figure 4: Reward function ablation test. Orange: no stability penalties, Black: no contact penalties, Green: no velocity penalties, Purple: no height rewards, Yellow: the default reward function. The default reward function with no shaping bonuses removed achieves the best performance, while a lack of stability penalties performs the worst.

- **Reproducibility**: Open-source implementation with clear environment specifications

6.2 Limitations and Future Work

- **Reward engineering overhead**: Iterative refinement was necessary to prevent exploit behaviors (e.g., indefinite forward motion)
- **Single task focus**: Current work addresses only forward locomotion; extension to diverse terrains requires further development
- **Sim-to-real gap**: No hardware validation conducted; transfer would require domain randomization and system identification
- **Sample efficiency**: While adequate, model-free PPO requires significantly more samples than model-based alternatives would in principle

Future work should explore: (1) domain randomization for sim-to-real transfer, (2) multi-task training across varied terrains, (3) comparison with hierarchical policies that decompose gait planning from execution, and (4) investigation of model-based methods when computational resources permit.

7 Conclusion

We demonstrated that careful reward shaping enables PPO to learn hexapod locomotion on a high-DoF robot without hand-coded gaits. Our key insight is that goal-directed rewards based on distance reduction are essential for task completion, whereas undirected forward motion rewards can lead to policies that move indefinitely without reaching goals. The contact-aware penalty structure successfully prevents crawling behaviors by explicitly checking contact locations. While our approach requires manual reward engineering, it provides a practical and interpretable solution for learning complex locomotion under computational constraints.

References

[1] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[2] M. Schilling, K. Konen, F. W. Ohl, and T. Korthals, "Decentralized deep reinforcement learning for a distributed and adaptive locomotion controller of a

hexapod robot," 2020. [Online]. Available: https://arxiv.org/abs/2005.11164

[3] A. S. Lele, Y. Fang, J. Ting, and A. Raychowdhury, "Learning to walk: Spike based reinforcement learning for hexapod robot central pattern generation," 2020. [Online]. Available: https://arxiv.org/abs/2003.10026

[4] W. Ouyang, H. Chi, J. Pang, W. Liang, and Q. Ren, "Adaptive locomotion control of a hexapod robot via bio-inspired learning," *Frontiers in Neurorobotics*, vol. Volume 15 - 2021, 2021. [Online]. Available: https://doi.org/10.3389/fnbot.2021.627157

[5] Z. Qiu, W. Wei, and X. Liu, "Adaptive gait generation for hexapod robots based on reinforcement learning and hierarchical framework," *Actuators*, vol. 12, no. 2, 2023. [Online]. Available: https://www.mdpi.com/2076-0825/12/2/75

[6] Y.-H. Tsai, S. Saeedvand, and J. Baltes, "A deep reinforcement learning algorithm for objects balance control with hexapod robot," in *2025 10th International Conference on Control and Robotics Engineering (ICCRE)*, 2025, pp. 34–39.

[7] T. Qu, D. Li, A. Zakhor, W. Yu, and T. Zhang, "Versatile locomotion skills for hexapod robots," 2024. [Online]. Available: https://arxiv.org/abs/2412.10628

[8] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," 2023. [Online]. Available: https://arxiv.org/abs/2309.14341

[9] H. Fu, K. Tang, P. Li, W. Zhang, X. Wang, G. Deng, T. Wang, and C. Chen, "Deep reinforcement learning for multi-contact motion planning of hexapod robots," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Z.-H. Zhou, Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2021, pp. 2381–2388, main Track. [Online]. Available: https://doi.org/10.24963/ijcai.2021/328

[10] J. Wu, G. Xin, C. Qi, and Y. Xue, "Learning robust and agile legged locomotion using adversarial motion priors," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4975–4982, 2023.

[11] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," 2024. [Online]. Available: https://arxiv.org/abs/2301.04104

[12] P. Wu, A. Escontrela, D. Hafner, K. Goldberg, and P. Abbeel, "Daydreamer: World models for physical robot learning," 2022. [Online]. Available: https://arxiv.org/abs/2206.14176

[13] 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*. PMLR, 2022, pp. 91–100.