Robot Parkour Enhancement and Sim2sim Deployment

Xiangchen Tian
Tsinghua University
Beijing, China
txc23@mails.tsinghua.edu.cn

Ruicheng Li
Tsinghua University
Beijing, China
lirc23@mails.tsinghua.edu.cn

Mengjie Zhao
Tsinghua University
Beijing, China
zhaomj23@mails.tsinghua.edu.cn

Zhikai Qin
Tsinghua University
Beijing, China
qinzk23@mails.tsinghua.edu.cn

Abstract

This project addresses quadruped parkour learning, where a low-cost legged robot must autonomously perform diverse locomotion skills—such as climbing, leaping, and tilting—in complex environments using only onboard perception. We reproduce the Robot Parkour Learning (RPL) framework in Isaac Gym and identify leaping over wide gaps as a key performance bottleneck. To address this, we extend the robot's perceptual horizon and adjust the reward's energy constraints, significantly improving leap success rates. We further evaluate sim-to-sim transfer to MuJoCo and analyze performance degradation caused by perception misalignment and terrain distribution shifts. Our work reveals key limitations in current visual control policies and offers insights into enhancing policy robustness and cross-domain generalization.

1 Introduction

Legged robots have long been envisioned as versatile agents capable of navigating complex and dynamic environments, much like animals in nature. However, achieving such agility and adaptability in robotic systems remains a significant challenge in robotics and artificial intelligence. Recent advancements in deep reinforcement learning (DRL) have shown promise in enabling robots to learn complex locomotion skills, including dynamic maneuvers and obstacle traversal. In this context, parkour—a discipline involving running, jumping, and climbing over obstacles—emerges as a quintessential benchmark for evaluating the agility and adaptability of legged robots.

The Quadruped Parkour Learning task, presents a unique opportunity to push the boundaries of legged robot locomotion. The goal is to train a quadruped robot, the Unitree Go2, to perform dynamic parkour maneuvers in visually diverse and complex environments. This task requires the robot to learn a set of distinct locomotion skills, including climbing, leaping, crawling, tilting, and running, and subsequently distill these skills into a single visual policy. The challenge lies not only in the complexity of the individual skills but also in the seamless integration and generalization of these skills across various environments.

In this project, we aim to reproduce and improve upon the state-of-the-art Robot Parkour Learning (RPL) framework[25] using the Isaac Gym simulator. The original RPL framework employs a two-stage reinforcement learning approach to train specialized skill policies and then distills these policies into a unified vision-based policy. While this framework demonstrates impressive results in

Preprint. Under review.

simulation, it also reveals limitations in certain scenarios, particularly in the leap skill when crossing wide gaps. Our work focuses on addressing these limitations and enhancing the overall performance, generalization, and robustness of the learned policies.

Our contributions can be summarized as follows:

- Reproduction of the RPL Framework: We successfully reproduce the original RPL framework in the Isaac Gym environment, achieving comparable performance in various parkour skills.
- Enhancement of Leap Skill Performance: Through targeted improvements in observation modeling and reward structure, we significantly enhance the robot's ability to leap over wide gaps, achieving higher success rates in challenging scenarios.
- Policy Deployment and Perception Alignment: We transfer the learned policies to the MuJoCo environment, addressing challenges in perception alignment and demonstrating the robustness and generalization capability of our policies across different simulators.

Through these contributions, we aim to advance the state of the art in legged robot parkour learning and provide insights into improving the performance and generalization of reinforcement learning policies in complex locomotion tasks.

2 Related Work

Model-based Control Strategies Model-based control strategies have driven remarkable advancements in agile robotic locomotion. In recent years, learning-based approaches have expanded the repertoire of agile behaviors, encompassing high-speed running[4, 8, 15], rapid recovery from random postures to standing[7, 23, 22, 13], jumping[11, 14], stair climbing[21, 10, 16, 1, 24, 12], climbing over obstacles[20], stepping-stone traversal[1],bipedal standing on rear legs[22], opening doors[3], moving with damaged parts[17], catching flying objects[2], playing football/soccer[6]. However, most of these skills either operate in sensor-agnostic modes or rely on explicit state estimation, with specialized designs for individual tasks.

Sim2sim/real Transfer Recent advances in sim-to-sim (sim2sim) transfer have demonstrated promising approaches to bridge the reality gap in robotic learning. Humanoid-Gym[5], an open-source framework that leverages domain randomization and high-fidelity simulators like MuJoCo to facilitate efficient sim2sim validation before real-world deployment, particularly for complex humanoid robots. Building on this, recent papers[9] explored sim2sim transfer between abstract topological environments and continuous 3D spaces in vision-and-language navigation, achieving a 12% improvement in success rate while analyzing the performance gap between different simulation paradigms. Meanwhile, AdaptSim[18] introduced a task-driven adaptation framework that meta-learns simulation parameter distributions to optimize task performance during sim2sim transfer, showing 2x data efficiency compared to traditional system identification methods.

3 Methods

3.1 Problem Formulation

Legged robot parkour learning represents an advanced challenge in autonomous locomotion. The key problem we address is: how to enable low-cost quadruped robots to autonomously learn and execute a wide range of vision-based parkour skills in diverse and unstructured environments.

We formulate the parkour learning task as a Markov Decision Process. Let $\mathcal{O}$ denote the set of obstacle configurations and $\pi(a_t|s_t, o_t)$ be the policy that outputs action a_t at time t given robot state s_t and observation o_t. The robot state includes proprioceptive signals such as joint angles, velocities, and base motion, while the observation o_t consists of egocentric depth images captured by the onboard sensor.

The goal is to learn a single end-to-end policy π that can operate from raw onboard sensory inputs and generalize across multiple parkour skills.

3.2 Parkour Skill Learning and Distillation in Isaac Gym

To develop an agile parkour system for quadruped robots navigating complex obstacle courses, we first replicated the two-stage reinforcement learning method proposed in the Robot Parkour Learning [25] paper within the Isaac Gym simulation environment. This unified framework automatically generates five parkour skills—climbing, leaping, crawling, tilting, and running—thereby avoiding the need for separate training schemes for each skill, which is common in prior work.

3.2.1 Two-Stage RL for Individual Parkour Skills

RL Pre-training with Soft Dynamics Constraints. In the RL Pre-training Stage, we introduced soft dynamics constraints and an automatic curriculum learning mechanism. Unlike traditional hard constraints that often lead to local optima, this stage allowed the robot to partially penetrate obstacles. By designing a penetration penalty reward:

$$r_{penetrate} = -\sum_{p}(\alpha_5 \cdot \mathbf{1}[p] + \alpha_6 \cdot d(p)) \cdot v_x,$$

where p represents the penetration points, α_5 and α_6 are hyperparameters, $d(p)$ is the penetration depth, and v_x is the forward velocity, we encouraged the robot to minimize penetrations while still learning to navigate obstacles. This approach enabled the robot to explore various parkour skills without being overly constrained by hard dynamics limits. After further integrating it with an adaptive curriculum, the robot gradually learned to overcome obstacles while minimizing penetrations and mechanical energy consumption.

Pre-training utilized the PPO algorithm, with a total reward function:

$$r = r_{skill} + r_{penetrate},$$

where r_{skill} is the general skill reward, which includes a forward reward $r_{forward}$, an energy reward r_{energy}, and an alive bonus r_{alive}. It motivates the robot to move forward efficiently and conserve mechanical energy.

RL Fine-tuning with Hard Dynamics Constraints. For the RL Fine-tuning Stage, the pre-trained policies were fine-tuned under realistic hard dynamics constraints, disallowing any penetration between the robot and obstacles. This stage also employed the PPO algorithm, but solely with the general skill reward r_{skill}. Each specialized skill policy (e.g., π_{climb}, π_{leap}) was trained independently. Notably, the running skill, which does not involve obstacles, was trained directly under hard dynamics constraints, bypassing the pre-training stage. Each specialized skill policy was parameterized as a Gated Recurrent Unit (GRU), taking as input proprioception information, the previous action, privileged visual information (such as distance to obstacle, height, width, and obstacle type), and privileged physics information.

3.2.2 Distillation to a Unified Vision-Based Policy

Following the learning of individual specialized skills, we employed the **DAgger (Dataset Aggregation)** [19] algorithm to distill these five distinct specialized skill policies into a single, vision-based parkour policy $\pi_{parkour}$. This distilled policy relies solely on onboard sensing (proprioception and depth images) for control, enabling robust deployment in real-world scenarios and autonomous switching between parkour skills based on environmental perception.

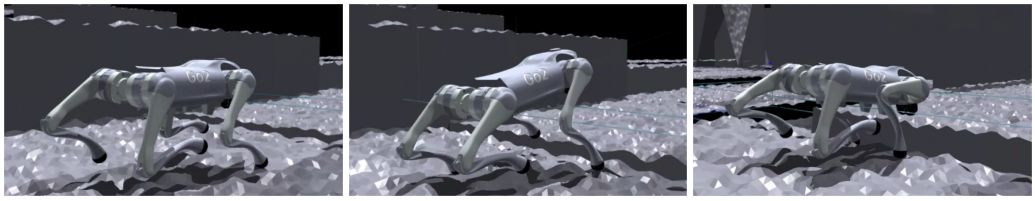
The distillation process was conducted in a simulated terrain with various obstacle types, where specialized skill policies acted as experts guiding $\pi_{parkour}$'s learning. Depth images were processed by a small CNN to extract features, which were then combined with proprioception and the previous action as inputs to $\pi_{parkour}$. This unified policy allows the robot to autonomously perceive obstacles and switch between skills in a fully end-to-end fashion, without relying on privileged information unavailable at test time.

3.2.3 Improvement on Leap Skill Performance

During our empirical evaluation of the original parkour learning system, we observed a notable performance bottleneck in the robot's ability to leap over wide gaps. Specifically, the field-trained

policies exhibited limited success when the gap width exceeded 0.5 meters. This weakness was inherited by the distilled vision-based policy, suggesting that the root cause may lie in the underlying skill policies learned in the field stage.

Upon closer analysis of the robot's behavior in various leap scenarios, we found that when the gap width was relatively small (e.g., $\leq 0.45\ m$), the robot would correctly initiate a jumping maneuver near the edge of the gap. However, as the gap width increased (e.g., $> 0.45\ m$), this behavior transitioned into a markedly different pattern. Instead of leaping forward, the robot tended to tilt downward and drop into the gap in a manner resembling the "`down`" behavior (descend from a platform) learned for a different skill. Figure 1 illustrates this phenomenon.



(a) moving forward (b) leap behavior at $0.3\ m$ gap (c) down behavior at $0.6\ m$ gap

Figure 1: Illustration of the robot's behavior in different scenarios: (a) moving forward, (b) leaping over a $0.3\ m$ gap, and (c) descending into a $0.6\ m$ gap. The robot's behavior transitions from a leap to a downward descent as the gap width increases.

Limited Perception Range of Privileged Observation Causes Misclassification. We hypothesized that this misclassification might stem from the limited spatial range of the privileged observation used during skill training. In the original configuration, the robot's forward perception along the forehead direction was restricted to 1.51 meters. This constraint could potentially prevent the policy from adequately distinguishing between a slightly wider gap and a descending platform, especially when approaching the gap at a moderate speed.

To test this hypothesis, we increased the perceptual range of the privileged visual input on the forward direction from $1.51\ m$ to $3.51\ m$. This modification enabled the policy to capture more distant obstacle features, improving its ability to distinguish large gaps from drop-offs in time to initiate an appropriate leap. Experimental results showed significant improvement in success rates for the $0.5\ m$ and $0.6\ m$ gap scenarios with this change, and the jump behavior emerged as expected.

Energy Penalty Adjustment for Enhanced Jump Ability. Despite this enhancement, the performance on $0.6\ m$ and $0.7\ m$ gaps remained suboptimal. Though the robot does try to jump, it often fails due to insufficient height and momentum, resulting in collisions with the gap edges or falling into the gap. We attributed this to the overly conservative reward function, particularly the energy penalty term that may have discouraged the robot from using powerful motions necessary for high-effort skills.

To address this, we fine-tuned the reward function by decreasing the weight of the energy cost term and simultaneously increasing the penalty for failure to complete the leap. This encouraged the robot to use more dynamic and energy-intensive actions while still being penalized for collisions or falling into the gap. This modification further brings a dramatic increase on the $0.6\ m$ scenario and showed positive trends for the $0.7\ m$ gap.

Overall, this two-step improvement—expanding the perception range and rebalancing the reward function—demonstrates that even with the original training pipeline, targeted adjustments to the observation model and reward structure can significantly enhance specific skill capabilities within the parkour policy. The detailed result comparison is demonstrated in section 4.

3.3 Policy Deployment and Perception Alignment in MuJoCo

To validate the generalization capability and robustness of our learned policies, we directly deployed the Isaac Gym-trained policy onto a Unitree Go2 quadruped robot model within the MuJoCo simulation environment. A critical challenge during this phase was ensuring precise alignment

of perception inputs between Isaac Gym and MuJoCo, particularly for depth images and terrain height maps.

We thoroughly investigated the depth image acquisition pipeline in Isaac Gym. Raw GPU tensors were converted to PyTorch tensors via `gymtorch.wrap_tensor`, then subjected to a series of preprocessing steps: negative depth values were converted to positive, depth ranges were clipped and linearly normalized to $[0.0, 1.0]$, image edges were cropped (from $(120, 160)$ to $(108, 144)$), and finally, the images were resized to $(48, 64)$ with pixel values clamped between $[0, 1]$. These depth images were generated by an onboard camera whose handle was created via the `_create_onboard_camera` function, and images were acquired using `self.gym.get_camera_image_gpu_tensor`. We resolved issues related to camera creation in Isaac Gym, ensuring proper rendering in non-headless modes.

For terrain height maps, the process involved measuring heights at specific sample points around the robot. These sample points were transformed based on the robot's pose (especially yaw) and root states, with terrain data queried using `self.terrain.get_terrain_heights(points)`. We confirmed the role of the `self.cfg.horizontal_scale` parameter in converting world coordinates to height map indices and corrected visualization errors stemming from inverted axis orders in the height map plots.

In MuJoCo, we utilized the `heightmap.py` tool from the `gym_quadruped` repository to obtain similar height map information, employing raycasting via the `raycast_sensor` function. During deployment, we identified and resolved input issues caused by mismatched observation space dimensions in the configuration file (e.g., `num_obs` being excessively large) and occasional errors related to EGL rendering context release. Furthermore, we precisely aligned the position and orientation of the front-facing camera in MuJoCo to match Isaac Gym's camera perception range and field of view, thereby rectifying issues where nearby objects appeared transparent and depth maps were entirely black after deployment.

Through meticulous perception alignment and error troubleshooting, the Go2 robot successfully demonstrated robust parkour performance in challenging MuJoCo environments, validating the effectiveness of the Parkour policy and its transferability between different simulators.

4 Experiment Results

4.1 Reproducing the Parkour Baseline

We begin our experiment by reproducing the original parkour policy proposed in the Robot Parkour Learning framework. Following the original two-stage training pipeline and distillation strategy, we train the specialized skill policies in Isaac Gym and distill them into a unified vision-based policy. All the settings and hyperparameters are kept consistent with the original setup.

In our test environment, we evaluated the parkour policy across the five standard skill scenarios: `jump`, `leap`, `down`, `hurdle`, `tilted_ramp`, `stairsup`, `discrete_rect`, and `wave`. The results in simulation confirm the robustness and versatility of the baseline system, particularly in climbing and crawling tasks, where the policy achieves near-perfect success rates.

However, compared to the robustness and versatility in other tasks, the leap skill is relatively weak, both in the original field policy and the distilled vision-based policy. The experiment results are shown in the next section.

4.2 Improvement on Leap Skill Performance

Follow the method proposed in section 3, we evaluated three versions of the field policy, focusing on leap skill, across gap widths ranging from $0.4\ m$ to $0.7\ m$:

- Baseline: The original policy trained in the field stage, with perceptual range of $1.51\ m$.
- Observation-Enhanced: The policy with an increased perceptual range of $3.51\ m$.
- Observation-Enhanced + Reward-Tuned: The policy with both enhanced perception of $3.51\ m$ and a modified reward function.

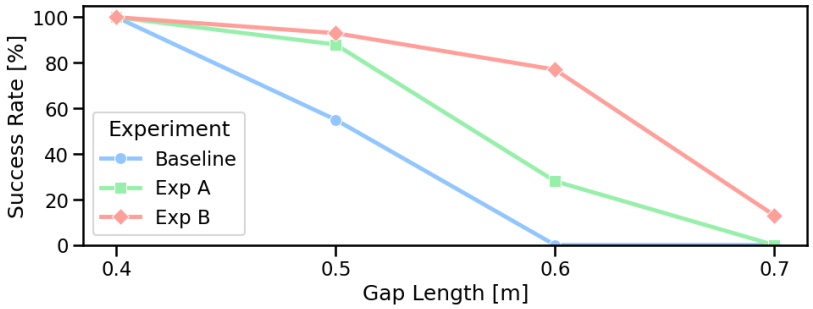


Figure 2: Performance comparison of the leap skill across different gap widths for three policy versions. Exp A is the Observation-Enhanced policy, and Exp B is the Observation-Enhanced + Reward-Tuned policy.

Gap Width (m)	Baseline Success Rate	Observation-Enhanced Success Rate	Reward-Tuned Success Rate
0.4	100%	100%	100%
0.5	50%	88.33%	93.33%
0.6	0%	28.33%	76.67%
0.7	0%	0%	13.33%

Table 1: Success rates of the leap skill across different gap widths for three policy versions.

The results are summarized in Figure 2 and Table 1 1

The Observation-Enhanced model significantly improves performance at $0.5\ m$ and $0.6\ m$ gaps, achieving 88.33% and 28.33% success rates respectively, compared to 50% and 0% in the baseline. But the success rate at $0.6\ m$ is still relatively low, with the intrinsic limitations of the original reward function preventing the robot from executing high-energy jumps. Further reward tuning boosts the success rate at $0.6\ m$ to 76.67%, and even enables partial success (13.3%) at $0.7\ m$, a case where the original policy always fails.

These results validate our hypothesis and demonstrate that targeted improvements in observation modeling and reward structure can significantly enhance the performance of the parkour skill in challenging scenarios. However, the policy still struggles with wide gaps such as $0.7\ m$, indicating that further refinements are needed to fully master this skill.

4.3 Policy Deployment and Perception Alignment in MuJoCo

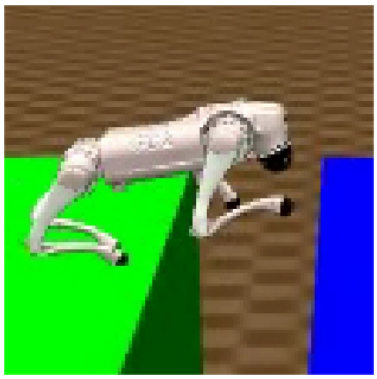
After training the field policy in Isaac Gym environment, we deploy the model in the MUJOCO environment, zero-shot, as shown in Figure 3a . We evaluated two versions of the field policy, focusing on leap skill, across gap widths ranging from $0.3\ m$ to $0.5\ m$:

- Baseline: The original policy trained in the field stage, with forward perceptual range of $1.51\ m$.
- Observation-Enhanced + Reward-Tuned: The policy with both enhanced forward perception of $3.51\ m$ and a modified reward function.

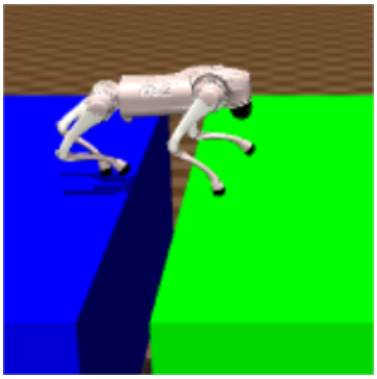
The results are summarized in Table 2

Gap Width (m)	Baseline Success Rate	Reward-Tuned Success Rate
0.3	100%	78.13%
0.4	93.33%	52%
0.5	0%	0%

Table 2: Success rates of the leap skill across different gap widths for three policy versions.



(a) Illustration of the field policy, leaping task.



(b) Illustration of the distilled policy, leaping task.

Figure 3: Deployment effect in MuJoCo.

After deployment, although the policy still remains the ability to leap, its performance has been significantly weakened. The base line has entirely failed to leap over the $0.5\ m$ gap, and the success rate to leap over the $0.4\ m$ gap has also decreased. Moreover, our enhanced policy also has lost its privilege after the zero-shot deployment, even with a decline rate while leaping over the gaps.

Such phenomenon may due to the improper alignment of parameters and simulation foundations between different simulation environments. However, the partly remained ability still shows the robustness of our trained policy.

Except for the field policy, we also deploy the distilled baseline policy in MuJoCo, see figure3b, the policy is able to walk normally, but its ability to leap is also weaker than that in the Isaac Gym environment.

We did the same experiment under this policy, the result is shown in table 3. More discussions about the sim2sim transfer will be conducted in section 5.

Gap Width (m)	Baseline Success Rate
0.3	54%
0.4	0%

Table 3: Success rates of the leap skill across different gap widths for three policy versions.

5 Discussion

Our experiments demonstrate clear progress in enhancing leap skill performance through observation range extension and reward tuning. However, the sim2sim deployment results remain suboptimal. This section discusses possible causes for this gap in performance, as well as additional directions for further improving the robot's leaping capability.

5.1 Challenges in Sim2Sim Deployment

Despite our efforts to align depth perception and observation preprocessing between Isaac Gym and MuJoCo, the leap success rate in MuJoCo significantly declined across all gap widths.

This discrepancy may arise from differences in the physics engines and their handling of contact dynamics, which can affect the robot's ability to execute effectively, especially in high-impact maneuvers like leaping.

Another key contributing factor may lie in the difference in terrain characteristics between the two simulators. In Isaac Gym, the parkour environments are rendered with slight terrain undulations and noise—deliberately introduced to increase diversity and robustness during training. However, in the initial MuJoCo deployment, the terrain was flat, with set friction parameters only, which altered the

local foot contact profiles. In response, we attempted to reproduce similar ground noise patterns in MuJoCo. Surprisingly, we found that the policy then failed even more catastrophically, exhibiting unstable behaviors or complete inaction.

To address these challenges, more robust training strategies are needed, such as more diverse domain randomization techniques or curriculum learning that gradually introduces variations in terrain features. Also, it is always a solution to align the physics parameters between the two simulators more precisely.

5.2 Insights into Leap Skill Improvement

We observe that for wider gaps, like $0.7\ m$ or larger, increasing the observation range will not bring corresponding performance improvements. Also, the leap initialization behavior becomes weak in this case. This indicates that limited observation is no longer the primary bottleneck, and the robot's motor capabilities are now the limiting factor. The lack of leap behavior is because the robot is unable to learn the leap skill for such difficult gaps at all during training.

A core factor underlying this limitation is the inherent trade-off between energy efficiency and task success embedded in the reward design. In reinforcement learning for legged locomotion, energy penalties are often critical for preventing erratic, unstable, or unrealistic motions. However, such constraints can also suppress the emergence of high-effort, high-impact behaviors like long-distance jumps. Our experiments show that loosening the energy constraint does lead to stronger actions and better performance, but this approach must be balanced carefully to avoid destabilizing overall locomotion or encouraging inefficient behavior in other tasks.

Beyond the energy design, several configurations during training may also have contributed to the bottleneck: First of all, the **expected running speed**. The policy is trained with a fixed target forward velocity. Though it is sufficient and suitable for most tasks, like climbing and jumping that do not require high speed, it is not enough for leaping over wide gaps. Therefore, increasing the expected pre-jump velocity or providing momentum-conditioning in the reward function specified for leaping task may allow the robot to leverage inertial dynamics more effectively. Secondly, the **Jump Height Expectations**. The current training method gives the robot a fixed target height by setting the height of fake obstacles in the environment. Though effective for relatively small gaps, it is clearly not proper for wide gaps. Hence it may be a further improvement to provide a dynamic target height based on the gap width, allowing the robot to adapt its jump height according to the specific challenge.

6 Conclusion

This study proposes an end-to-end visual control strategy for quadruped robot parkour, achieving multi-skill integration and environmental generalization through a two-stage RL framework and DAgger distillation algorithm. Core contributions include: 1) reproducing and improving the RPL framework in Isaac Gym, enhancing training efficiency through soft dynamics constraints and adaptive curriculum learning; 2) significantly improving the performance of the leaping skill in wide gap scenarios by expanding the perception range and tuning the reward function; 3) successfully transferring the policy to MuJoCo, verifying cross-simulator generalization ability.

This research provides a feasible solution for low-cost quadruped robots to perform autonomous parkour in complex environments, and the method can be extended to scenarios such as disaster relief and terrain exploration. Future research will focus on enhancing robustness in extreme scenarios, exploring hardware-software co-optimization with physical robots, and expanding the policy's adaptability to dynamic obstacles.

References

[1] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision, 2022.

[2] Benedek Forrai, Takahiro Miki, Daniel Gehrig, Marco Hutter, and Davide Scaramuzza. Event-based agile object catching with a quadrupedal robot, 2023.

[3] Zipeng Fu, Xuxin Cheng, and Deepak Pathak. Deep whole-body control: Learning a unified policy for manipulation and locomotion, 2022.

[4] Zipeng Fu, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Minimizing energy consumption leads to the emergence of gaits in legged robots, 2021.

[5] Xinyang Gu, Yen-Jen Wang, and Jianyu Chen. Humanoid-gym: Reinforcement learning for humanoid robot with zero-shot sim2real transfer. *arXiv preprint arXiv:2404.05695*, 2024.

[6] Tuomas Haarnoja, Ben Moran, Guy Lever, Sandy H. Huang, Dhruva Tirumala, Jan Humplik, Markus Wulfmeier, Saran Tunyasuvunakool, Noah Y. Siegel, Roland Hafner, Michael Bloesch, Kristian Hartikainen, Arunkumar Byravan, Leonard Hasenclever, Yuval Tassa, Fereshteh Sadeghi, Nathan Batchelor, Federico Casarini, Stefano Saliceti, Charles Game, Neil Sreendra, Kushal Patel, Marlon Gwira, Andrea Huber, Nicole Hurley, Francesco Nori, Raia Hadsell, and Nicolas Heess. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *Science Robotics*, 9(89), April 2024.

[7] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26), January 2019.

[8] Gwanghyeon Ji, Juhyeok Mun, Hyeongjun Kim, and Jemin 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, April 2022.

[9] Jacob Krantz and Stefan Lee. Sim-2-sim transfer for vision-and-language navigation in continuous environments. In *European conference on computer vision*, pages 588–603. Springer, 2022.

[10] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47), October 2020.

[11] Zhongyu Li, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Robust and versatile bipedal jumping control through reinforcement learning, 2023.

[12] Antonio Loquercio, Ashish Kumar, and Jitendra Malik. Learning visual locomotion with cross-modal supervision, 2022.

[13] Yuntao Ma, Farbod Farshidian, and Marco Hutter. Learning arm-assisted fall damage reduction and recovery for legged mobile manipulators, 2023.

[14] Gabriel B. Margolis, Tao Chen, Kartik Paigwar, Xiang Fu, Donghyun Kim, Sangbae Kim, and Pulkit Agrawal. Learning to jump from pixels, 2021.

[15] Gabriel B Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning, 2022.

[16] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62), January 2022.

[17] Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S. Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning, 2019.

[18] Allen Z Ren, Hongkai Dai, Benjamin Burchfiel, and Anirudha Majumdar. Adaptsim: Task-driven simulation adaptation for sim-to-real transfer. *arXiv preprint arXiv:2302.04903*, 2023.

[19] Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011.

[20] Nikita Rudin, David Hoeller, Marko Bjelonic, and Marco Hutter. Advanced skills by learning locomotion and local navigation end-to-end, 2022.

[21] Jonah Siekmann, Kevin Green, John Warila, Alan Fern, and Jonathan Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning, 2021.

[22] Laura Smith, J. Chase Kew, Tianyu Li, Linda Luu, Xue Bin Peng, Sehoon Ha, Jie Tan, and Sergey Levine. Learning and adapting agile locomotion skills by transferring experience, 2023.

[23] Chuanyu Yang, Kai Yuan, Qiuguo Zhu, Wanming Yu, and Zhibin Li. Multi-expert learning of adaptive legged locomotion. *Science Robotics*, 5(49), December 2020.

[24] Ruihan Yang, Ge Yang, and Xiaolong Wang. Neural volumetric memory for visual locomotion control, 2023.

[25] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher Atkeson, Soeren Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning, 2023.

Appendix

The group contribution is listed as follows:

- Lead the reproduction and improvement over the parkour learning baseline: Ruicheng Li, Mengjie Zhao
- Lead the deployment and sim2sim transfer in MuJoCo: Xiangchen Tian, Zhikai Qin
- Manuscript editing: Ruicheng Li, Mengjie Zhao, Xiangchen Tian, Zhikai Qin