Robot Parkour Learning

Anonymous Author(s)
Affiliation
Address
email

Figure 1: We present a framework for learning parkour skills on low-cost robots. Our end-to-end vision-based parkour learning system enable the robot to climb high obstacles, leap over large gaps, crawl beneath low barriers, squeeze through thin slits and run. Videos on the project website.

Abstract: Parkour is a grand challenge for legged locomotion that requires robots to overcome various obstacles rapidly in complex environments. Existing methods can generate either diverse but blind locomotion skills or vision-based but specialized skills by using reference animal data or complex rewards. However, *autonomous* parkour requires robots to learn generalizable skills that are both vision-based and diverse to perceive and react to various scenarios. In this work, we propose a system for learning a single end-to-end vision-based parkour policy of diverse parkour skills using a simple reward without any reference motion data. We develop a reinforcement learning method inspired by direct collocation to generate parkour skills, including climbing over high obstacles, leaping over large gaps, crawling beneath low barriers, squeezing through thin slits, and running. We distill these skills into a single vision-based parkour policy and transfer it to a quadrupedal robot using its egocentric depth camera. We demonstrate that our system can empower low-cost quadrupedal robots to autonomously select and execute appropriate parkour skills to traverse challenging environments in the real world. *We will fully open source our training code and hardware code upon publication.* Website: https://agile-locomotion.github.io

Keywords: Agile Locomotion, End-to-End Vision-Based Control, Sim-to-Real

1 Introduction

Humans and animals possess amazing athletic intelligence. Parkour is an examplar of athletic intelligence of many biological beings capable of moving swiftly and overcoming various obstacles

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

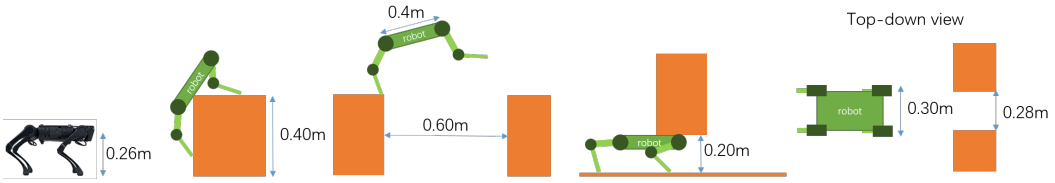

Figure 2: We illustrate the challenging obstacles that our system can solve, including climbing high obstacles of 0.40m (1.53x robot height), leap over large gaps of 0.60m (1.5x robot length), crawling beneath low barriers of 0.2m (0.76x robot height), squeezing through thin slits of 0.28m by tilting (less than the robot width).

in complex environments by running, climbing, and jumping [1]. Such agile and dynamic movements require real-time visual perception and memorization of surrounding environments [2, 3], tight coupling of perception and action [4, 5], and powerful limbs to negotiate barriers [6]. One of the grand challenges of robot locomotion is building autonomous parkour systems.

Boston Dynamics Atlas robots [7] have demonstrated stunning parkour skills. However, the massive engineering efforts needed for modeling the robot and its surrounding environments for predictive control and the high hardware cost prevent people from reproducing parkour behaviors given a reasonable budget. Recently, learning-based methods have shown robust performance on walking [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 12, 21, 22, 23, 24, 25, 26], climbing stairs [20, 27, 28, 29, 30, 31, 32], mimicking animals [33, 34, 35, 36, 37, 38] and legged mobile manipulation [39, 40] by learning a policy in simulation and transferring it to the real world while avoiding much costly engineering and design needed for robot-specific modeling. Can we leverage learning-based methods for robot parkour but only using low-cost hardware?

There are several challenges for robot parkour learning. First, learning diverse parkour skills (e.g. running, climbing, leaping, crawling, squeezing through, and etc) is challenging. Existing reinforcement learning works craft complex reward functions of many terms to elicit desirable behaviors of legged robots. Often each behavior requires manual tuning of the reward terms and hyper-parameters; thus these works are not scalable enough for principled generation of a wide range of agile parkour skills. In contrast, learning by directly imitating animals' motion capture data can circumvent tedious reward design and tuning [33, 41], but the lack of egocentric vision data and diverse animal MoCap skills prevents the robots from learning diverse agile skills and autonomously selecting skills by perceiving environment conditions. Second, obstacles can be challenging for low-cost robots of small sizes, as illustrated in Figure 2. Second, beyond the challenge of learning diverse skills, visual perception is dynamical and laggy during high-speed locomotion. For example, when a robot moves at 1m/s, a short 0.2 second of signal communication delay will cause a perception discrepancy of 0.2m (7.9 inches). Existing learning-based methods have not demonstrated effective high-speed agile locomotion. Lastly, parkour drives the electric motors to their maximum capacity, so proactive measures to mitigate potential damage to the motors must be included in the system.

This paper introduces a robot parkour learning system for low-cost quadrupedal robots that can perform various parkour skills, such as climbing over high obstacles, leaping over large gaps, crawling beneath low barriers, squeezing through thin slits, and running. Our reinforcement learning method is inspired by direct collocation and consists of two simulated training stages: RL pre-training with soft dynamics constraints and RL fine-tuning with hard dynamics constraints. In the RL pre-training stage, we allow robots to penetrate obstacles using an automatic curriculum that enforces soft dynamics constraints. This encourages robots to gradually learn to overcome these obstacles while minimizing penetrations. In the RL fine-tuning stage, we enforce all dynamics constraints and fine-tune the behaviors learned in the pre-training stage with realistic dynamics. In both stages, we only use a simple reward function that motivates robots to move forward while conserving mechanical energy. After each individual parkour skill is learned, we use distill DAgger [42] them into a single vision-based parkour policy that can be deployed to a legged robot using only onboard perception and computation power.

The main contributions of this paper include:

- **a system for robot parkour learning**, validating low-cost robots powered by our system can also parkour in the real world;
- **a two-stage RL method** for overcoming hard exploration problems, involving a pre-training stage with soft dynamics constraints and a fine-tuning stage with hard dynamics constraints;

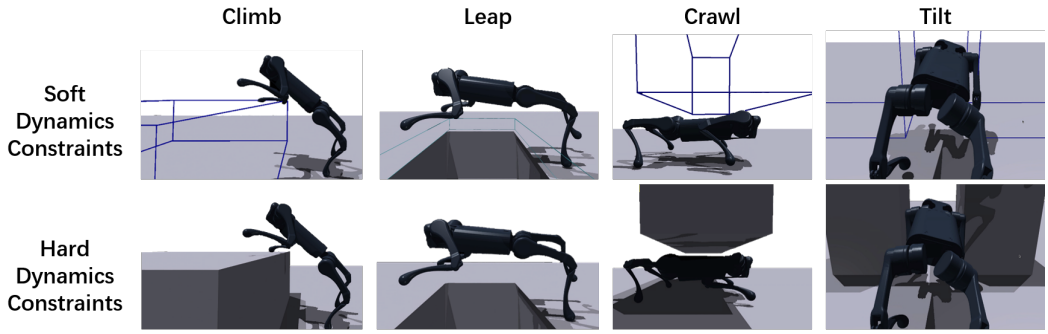

Figure 3: Soft dynamics constraints and hard dynamics constraints for each skill. Given soft dynamics constraints, the obstacles are penetrable.

- **extensive experiments in simulation and the real world** showing that our parkour policy enables low-cost quadrupedal robots to autonomously select and execute appropriate parkour skills to traverse challenging environments in the open world using only onboard computation, onboard visual sensing and onboard power, including climbing high obstacles of 0.40m (1.53x robot height), leap over large gaps of 0.60m (1.5x robot length), crawling beneath low barriers of 0.2m (0.76x robot height), squeezing through thin slits of 0.28m by tilting (less than the robot width), and running.

2 Related Work

Agile Locomotion. Model-based control has achieved much success in agile locomotion, from MIT Cheetah robots and A1 robots jumping over or onto obstacles of various heights [43, 44, 45], ETH StarlETH robots jumping vertically [46], CMU Unified Snake robots climbing trees [47], X-RHex robots self-righting using tails [48], ANYmal ALMA robots opening doors [49], ATRIAS robots walking over stepping stones [50, 51], Marc Raibert's One-Legged Hopping Machine [52], and Boston Dynamics Atlas' parkour skills [7]. Recently, learning-based methods have also demonstrated various agile locomotion capabilities, including high-speed running [53, 16, 54, 34], resetting to the standing pose from random states [11, 37, 15, 55], jumping [56, 57, 58], climbing stairs [20, 10, 27, 28, 29, 31, 32], climbing over obstacles [59], walking on stepping stones [28], back-flipping [60], quadrupedal standing up on rear legs [41], opening doors [39, 61, 62, 63], moving with damaged parts [64], catching flying objects [65], balancing using a tail [66], playing football/soccer [67, 68, 69, 70], weaving through poles [71] and climbing ramps [71]. Most of these skills are blind or rely on state estimation, and specialized methods are designed for these individual skills. In contrast, we focus on learning a single end-to-end vision-based parkour policy for a wide range of parkour skills in one system.

Vision-Based Locomotion. Classical modular methods rely on decoupled visual perception and control pipelines, where the elevation maps [72, 73, 74, 75, 76, 77, 78], traversability maps [79, 80, 81, 82], or state estimators [83, 84, 85, 86, 87, 88, 89, 90, 91, 92] are constructed as intermediate representations for downstream foothold planning, path planning and control [93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105]. Recently, end-to-end learning-based methods have also incorporated visual information into locomotion, where visual perception is performed using depth sensing [28, 58, 30], elevation maps [27, 106, 107, 108, 109], lidar scans [110], RGB images [31], event cameras [65] or learned neural spaces [29, 32], but none have demonstrated effective high-speed agile locomotion.

3 Robot Parkour Learning Systems

Our goal is to build an end-to-end parkour system that directly uses raw onboard depth sensing and proprioception to control every joint of a low-cost robot to perform various agile parkour skills, such as climbing over high obstacles, leaping over large gaps, crawling beneath low barriers, squeezing through thin slits, and running. Unlike prior work where different methods and training schemes are used for different locomotion skills, we aim to generate these five parkour skills automatically and systemically. To achieve this, we develop a two-stage reinforcement learning method that is inspired by direct collocation to learn these parkour skills under the same framework. In the RL

pre-training stage, we allow robots to penetrate obstacles using an automatic curriculum that enforces soft dynamics constraints. We encourage robots to gradually learn to overcome these obstacles while minimizing penetrations and mechanical energy. In the RL fine-tuning stage, we fine-tune the pre-trained behaviors with realistic dynamics. In both stages, we only use a simple reward function that motivates robots to move forward while conserving mechanical energy. After each individual parkour skill is learned, we use DAgger [42] to distill them into a single vision-based parkour policy that can be deployed. For robust sim-to-real deployment on a low-cost robot, we employ several pre-processing techniques for the depth images, calibrate onboard visual delays, and enforce proactive measures for motor safety.

3.1 Parkour Skills Learning

Since depth sensing is costly to simulate and directly training RL on visual data is not always stable, we use privileged visual information about the environments to help RL to generate specialized parkour skills in simulation. The privileged visual information includes the distance from the robot's current position to the obstacle in front of the robot, the height of the obstacle, the width of the obstacle, and a 4-dim one-hot category representing the four types of obstacles. We formulate each specialized skill policy as a gated recurrent neural network (GRU [111]). The inputs to a policy other than the recurrent latent state are proprioception $s_t^{\text{proprio}} \in \mathbb{R}^{29}$ (row, pitch, base angular velocities, positions and velocities of joints), last action $a_{t-1} \in \mathbb{R}^{12}$, the privileged visual information e_t^{vis}, and the privileged physics information e_t^{phy}. We use a similar approach to prior work [8, 10] to sample physics properties like terrain friction, center of mass of the robot base, motor strength and etc to enable domain adaptation from simulation to the real world. The policy outputs the target joint positions $a_t \in \mathbb{R}^{12}$.

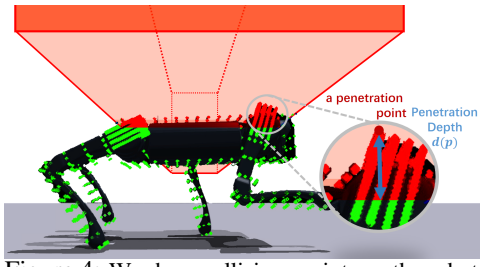

Figure 4: We show collisions points on the robot. Collision points that penetrate obstacles are in red.

We train all the specialized skill policies $\pi_{\text{climb}}, \pi_{\text{leap}}, \pi_{\text{crawl}}, \pi_{\text{tilt}}, \pi_{\text{run}}$ on corresponding terrains shown in Figure 3 using the same reward structure. We use the formulation of minimizing mechanical energy in [34] to derive a general task reward r_{task} suitable for generating all skills with natural motions, which only consists of three parts, a forward reward r_{forward}, an energy reward r_{energy} and an alive bonus r_{alive}:

$$r_{\text{task}} = r_{\text{forward}} + r_{\text{energy}} + r_{\text{alive}},$$

where

$$r_{\text{forward}} = -\alpha_1 * |v_x - v_x^{\text{target}}| - \alpha_2 * |v_y|^2 + \alpha_3 * e^{-|\omega_{\text{yaw}}|},$$
$$r_{\text{energy}} = -\alpha_4 * \sum_{j \in \text{joints}} |\tau_j \dot{q}_j|^2, \quad r_{\text{alive}} = 2.$$

Measured at every time step, v_x is forward base linear velocity, v_x^{target} is the target speed, v_y is the lateral base linear velocity, ω_{yaw} is the base angular yaw velocity, τ_j is the torque at joint j, ω_{yaw} is the joint velocity at at joint j, and α are hyperparameters. We set the target speed for all skills to around 1 m/s. We use the second power of energy consumption at each joint to reduce both average and variance of energy consumption across all joints. See supplementary for all the hyperparameters.

RL Pre-training with Soft Dynamics Constraints. As illustrated in Figure 2, the limited size of the robot compared with the large size of obstacles prevents generic RL algorithms from effectively finding policies that can overcoming these obstacles. Inspired direct collocation with soft constraints, we propose to use soft dynamics constraints to solve this hard exploration problem. Shown in Figure 3, we set the obstacles to be penetrable so the robot can violate the physical dynamics in the simulation by directly go through the obstacles without get stuck before the obstacles as a result of local minima of RL training with hard dynamics constraints. Similar to the Lagrangian formulation of direct collocation [112], we develop a penetration reward $r_{\text{penetrate}}$ to

Skill	Obstacle Properties	Ranges ($[l_{\text{easy}}, l_{\text{hard}}]$)
Climb	obstacle height	[0.2, 0.45]
Leap	gap length	[0.2, 0.8]
Crawl	clearance	[0.32, 0.22]
Tilt	path width	[0.32, 0.28]

Table 1: Ranges for obstacle properties for each skill during training, measured in meters.

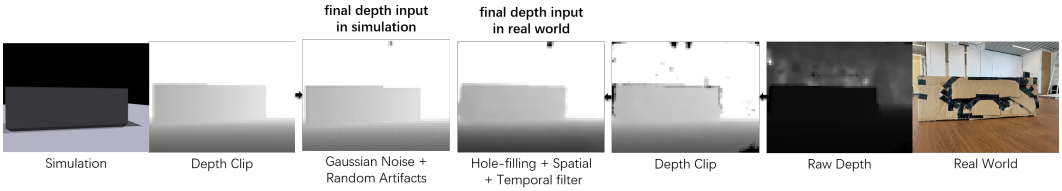

Figure 5: We bridge the gap between depth image in simulation and real world. We use clipping, Gaussian noise and artifacts in the simulation, and clipping, hole-filling filter and spatial/temporal filter in the real world.

gradually enforce the dynamics constraints and an automatic curriculum that adaptively adjusts the difficulty of obstacles. Shown in Figure 4, to measure the degree of dynamics constraints' violation, we sample collision points on the collision bodies of the robot in order to measure the volume and the depth of penetration. Since the hips and shoulders of the robot contain all the motors, we sample more collision points around these regions to enforce stronger dynamics constraints, encouraging fewer collisions of these vulnerable body parts in the real world. Denote a collision point on the collision bodies as p, an indicator function of whether p violates the soft dynamics constraints or not as $\mathbb{1}[p]$, and the minimal distance of p to the penetrated obstacle surface as $d(p)$. The volume of penetration can be approximated by the sum of $\mathbb{1}[p]$ over all the collision points, and the average depth of penetration can be approximated by the sum of $d(p)$. In Figure 4, the collisions points violating the soft dynamics constraints ($\mathbb{1}[p] = 1$) are in red, and those with $\mathbb{1}[p] = 0$ are in green. Concretely, the penetration reward is

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

where α_5 and α_6 are two fixed constants. We multiply both the penetration volume and the penetration depth with forward base velocity v_x to prevent the robot from exploiting the penetration reward by sprinting through the obstacles to avoid high cumulative penalty over time. In addition, we implement an automatic curriculum that adaptively adjust the difficulty of the obstacles after a reset based on the performance of individual robots simulated in parallel in simulation. We first calculate the performance of a robot based on its penetration reward averaged over the previous episode before the reset. If the penetration reward is over a threshold, we increase the difficulty score (s) of obstacles that the robot will face by one unit (0.05); if lower, then we decrease it by one unit. Every robot starts with a difficulty score 0 and the maximum difficulty score is 1. We set the obstacle propriety for the robot based on its difficulty score by $(1 - s) * l_{\text{easy}} + s * l_{\text{hard}}$, where l_{easy} and l_{hard} are the two limits of the ranges of obstacles properties corresponding to different parkour skills (shown in Table 1. We pre-train the specialized parkour skills with soft dynamics constraints using PPO [113] with the sum of the general task reward and the penetration reward $r_{\text{task}} + r_{\text{penetrate}}$.

RL Fine-tuning with Hard Dynamics Constraints. After the pre-training stage of RL is near convergence, we fine-tune every specialized parkour skill policy on the realistic hard dynamics constraints (shown in Figure 3); hence, no penetrations between the robots and obstacles are possible at the second stage of RL. We use PPO to fine-tune the specialized skills using only the general task reward r_{task}. We randomly sample obstacle properties from the ranges listed in Table 1 during fine-tuning. Since the running skill is trained on terrains without obstacles, we directly train the running skill with hard dynamics constraints and skip the RL pre-training stage with soft dynamics constraints.

3.2 Learning a Single Parkour Policy by Distillation

The learned specialized parkour skills are five policies that use both the privileged visual information e_t^{vis}, and the privileged physics information e_t^{phy}. However, the ground-truth privilege information is only available in the simulation but not in the real world. Furthermore, each specialized policy can only execute one skill but cannot autonomously execute and switch between different parkour skills based on visual perception of the environments. We propose to use DAgger [42] to distill a single vision-based parkour policy π_{parkour} using only onboard sensing from the five specialized skill policies $\pi_{\text{climb}}, \pi_{\text{leap}}, \pi_{\text{crawl}}, \pi_{\text{tilt}}, \pi_{\text{run}}$. We randomly sample obstacles types and properties from Table 1 to form a simulation terrain consisting of 40 tracks and 20 obstacles on each track. Since we have full knowledge of the type of obstacle related to every state s_t, we can assign the corresponding specialized skill policy $\pi_{s_t}^{\text{specialized}}$ to teach the parkour policy how to act at a state. For example, we

	Success Rate (%) ↑					Average Distance (m) ↑				
	Climb	Leap	Crawl	Tilt	Run	Climb	Leap	Crawl	Tilt	Run
Blind	0	0	13	0	100	1.53	1.86	2.01	1.62	3.6
MLP	0	1	63	43	100	1.59	1.74	3.27	2.31	3.6
No Distill	0	0	73	0	100	1.57	1.75	2.76	1.86	3.6
RMA [8]	-	-	-	**74**	-	-	-	-	**2.7**	-
Ours (parkour policy)	**86**	**80**	**100**	73	100	**2.37**	**3.05**	**3.6**	2.68	3.6
Oracles w/o Soft Dyn	0	0	93	86	100	1.54	1.73	3.58	1.73	3.6
Oracles	95	82	100	100	100	3.60	3.59	3.6	2.78	3.6

Table 2: We test our method against several baselines and ablations in the simulation with a max distance of 3.6m. We measure the success rates and average distances of every skill averaged across 100 trials and three random seeds. Our parkour policy shows the best performance using only sensors that are available in the real world.

assign the climb policy π_{climb} to supervise the parkour policy given a high obstacle. We parameterize the policy as a GRU. The inputs except the latent state are the proprioception s_t^{proprio}, the previous action a_{t-1} and a latent embedding of the depth image I_t^{depth} processed by a small CNN. The distillation objective is

$$\arg\min_{\theta_{\text{parkour}}} \mathbb{E}_{s_t, a_t \sim \pi_{\text{parkour}}, sim} \left[D\left(\pi_{\text{parkour}}\left(s_t^{\text{proprio}}, a_{t-1}, I_t^{\text{depth}}\right), \pi_{s_t}^{\text{specialized}}\left(s_t^{\text{proprio}}, a_{t-1}, e_t^{\text{vis}}, e_t^{\text{phy}}\right)\right)\right],$$

where θ_{parkour} are the network parameters of the parkour policy, sim is the simulator with hard dynamics constraints, and D is the divergence function which is binary cross entropy loss for policy networks with tanh as the last layer. Both polices π_{parkour} and $\pi_{s_t}^{\text{specialized}}$ are stateful. More details of the parkour policy network are in the supplementary.

3.3 Sim-to-Real and Deployment

Although the distillation training in Section 3.2 can bridge the sim-to-real gap in physical dynamics properties such as terrain friction and mass properties of the robot [8, 10], we still need to address the sim-to-real gap in visual appearance between the rendered depth image in simulation and the onboard depth image taken by a depth camera in the real world. Shown in Figure 5, we apply pre-processing techniques to both the raw rendered depth image and the raw real-world depth image. We apply depth clipping, pixel-level Gaussian noise, and random artifacts to the raw rendered depth image, and apply depth clipping, hole filing, spatial smoothing and temporal smoothing to the raw real-world depth image.

The depth images in both simulation and real-world have a resolution of 48 * 64. Due to the limited computation power onboard, the refresh rate of onboard depth image is 20Hz. Our parkour policy operates at 50Hz in both simulation and the real world to enable agile locomotion skills, and asynchronously fetches the latest latent embedding of the depth image processed by a small CNN. The output actions of the policy are target joint positions which are converted in torques on the order of 1000Hz through a PD controller of $K_p = 50$ and $K_d = 1$. To ensure safe deployment, we apply a torque clipping of 25Nm by clipping target joint positions: $\text{clip}(q^{\text{target}}, (K_d * \dot{q} - 25)/K_p + q, (K_d * \dot{q} + 25)/K_p + q)$.

4 Experimental Results

Robot and Simulation Setup We use IsaacGym [114] as the simulator to train all the policies. To train the specialized parkour skills, we construct large environments consisting of 40 tracks and 20 obstacles on each track. The obstacles in each track have linearly increasing difficulties based on the obstacle property ranges in Table 1. Physics properties such as obstacle length and fractal noise scale are randomized across tracks. We use a Unitree A1 equipped with Nvidia Jetson NX for onboard computation and Intel RealSense D435 for onboard visual sensing. More details are in the supplementary.

Baselines and Ablations We compare our parkour policy with several baselines and ablations. The baselines include **Blind**, **RND** [115], **MLP** and **RMA** [8]. The ablations include **No Distill**,

Oracles w/o Soft Dyn. We also include **Oracles**, specialized parkour skills conditioned on priviledge information in simulation, for the completeness of the comparisons.

- **Blind**: a blind parkour policy baseline distilled from the specialized skills, implemented by setting depth images I^{depth} as zeros.
- **RND**: a RL exploration baseline method for training specialized skills with bonus rewards based on forward prediction errors. We train it without our RL pre-training on soft dynamics constraints.
- **MLP**: a MLP parkour policy baseline distilled from the specialized skills. Instead of using a GRU, it uses only the depth image, proprioception and previous action at one time step without any memory to output actions.
- **RMA**: a domain adaptation baseline that distills a parkour policy on a latent space of environment extrinsics instead on the action space.
- **No Distill**: an ablation training a vision-based parkour policy with GRU directly using PPO with our two-stage RL method but but skipping the distillation stage.
- **Oracles w/o Soft Dyn**: an ablation training specialized skill policies using priviledged information directly with hard dynamics constraints.
- **Oracles** (w/ Soft Dyn): our specialized skill policies using privileged information trained with our two-stage RL approach.

4.1 Simulation Experiments

Vision is crucial for learning parkour. We compare the Blind baseline with ours. Shown in Table 2, without depth sensing and relying only on proprioception, the distilled blind policy cannot complete any climbing, leaping or tilting trials and can only achieve a 13% success rate on crawling. This is expected, as vision enables sensing of the obstacles properties and prepare the robot for execute proper agile skills while approaching the obstacles.

RL pre-training with soft dynamics constraints enables parkour skills' learning. We compare the RND, Oracles w/o Soft Dyn and ours (Oracles w/ Soft Dyn), all trained using privileged information without the distillation stage. We aim to verify that if our method of RL pre-training with soft dynamics constraints efficiently solve the hard exploration problems incurred by the difficulties of the parkour skills and obstacles. In Figureřeffig:rnd, we measure the average success rates of each method averaged over 100 trials across all the parkour skills that require exploration including climbing, leaping, crawling and tilting. We trained using three random seeds for each method to measure the standard deviations. Our method using RL pre-training with soft dynamics constraints can achieve much faster learning progress and a better final success rate around 95%. We notice that RND struggles to learn meaningful behaviors with scenarios that require fine-grained maneveurs such as crawling through a thin slit, due to its tendency to reach states that are hard to be forward predicted. Both RND and Oracles w/o Soft Dyn cannot make any learning progress on climbing and leaping, the two most difficult parkour skills. More plots showing the success rates for each skill separately are in the supplementary.

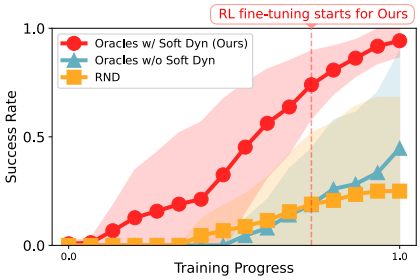

Figure 6: Comparison of specialized oracles trained with soft dynamics constraints with baselines averaged across every skill and three trials.

Recurrent networks enable parkour skills requiring memories. We compare the MLP baseline with ours using a GRU to parameterize the vision-based parkour policy. Shown in Table 2, the MLP baseline cannot learn the climbing and leaping skills and achieve much lower performance on crawling and tilting. Both climbing and leap requires the robot to hold a short-term memory of the past visual perceptions. For example, during climbing when the robot has its front legs on the obstacles, it still needs memory about the spatial dimensions of the obstacle captured in past depth images to control the rear legs to complete the climbing.

Distillation is effective for Sim2Real. We compare the RMA baseline and the No Distill baseline with ours. Although RMA can achieve similar performance on one skill that it is trained on (tilting), RMA directly copies and fixes the network parameters of the MLP processes the latent embeddings of the backbone GRU that outputs actions from the special skill trained on privilege information to the distilled policy, so it cannot distill multiple specialized skill policies, which have different MLP

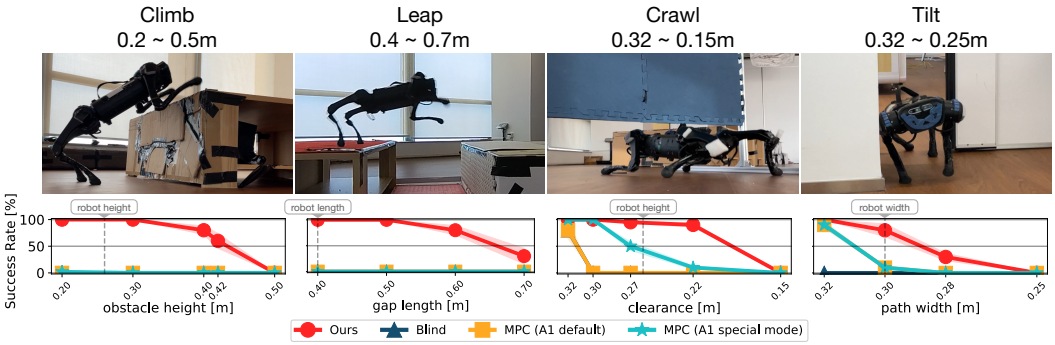

Figure 7: Real-world indoor quantitative experiments. Our parkour policy can achieve the best performance, compared with a blind policy and built-in A1 controllers. We control A1 special mode by teleoperating the robot lower down or tilt the body during crawling and tilt respectively.

parameters, into one parkour policy. No Distill cannot learn climbing, leaping and tilting due to the complexity of training directly from visual observations without privileged information.

4.2 Real-World Experiments

Emergent Re-trying Behaviors during Climbing Our parkour policy has emergent re-trying behaviors in the real world. When trying to overcoming a high obstacle but fails at the first trial, the robot will push itself away from the obstacle to prepare space for a second re-trying attempt. Although we do not program such re-trying behaviors, they nicely emerge out of learning with simple rewards. This behavior is also observed in simulation.

Indoor Quantitative Experiments Shown in Figure 1, we test our parkour policy in a constructed parkour trial consisting of crawling, climbing, and leaping in sequential. We also conduct quantitative indoor experiments in the real world on the A1 robot. In Figure 7, we compare our vision-based parkour policy, with Blind, MPC (A1 default controller) and MPC (A1 special mode). We show the success rates of each method in every task under varying difficulties averaged over 10 trials each. We change the task difficulty by modify the key obstacle properties, such as obstacle heights for climbing and gap length for leaping. In A1 special mode, we directly teleoperate the robot to change its state, such as lower down the body during crawling. We observe that our parkour policy can enable the robot to climb obstacles as high as 0.40m (1.53x robot height) with 80% success rate, to leap over gaps as large as of 0.60m (1.5x robot length) with 80% success rate, to crawl beneath barriers as low as of 0.2m (0.76x robot height) with 90% success rate, and to squeeze through thin slits of 0.28m by tilting (less than the robot width). Our method has the best performance across all skills. Please refer to our project website for indoor experiment videos.

Outdoor Experiments We test our robot outside of the lab in the outdoor environments. Shown in Figure 1, we observe that the robot powered by our parkour policy can complete a wide range of agile parkour skills. It can leap over two disconnect stone stools by the river with a 0.6m wide gap. It can continuously climb several stairs of 0.42m high each. It can crawl beneath a camping cart as well as handle the slippery grass terrain. Please refer to our project website for outdoor experiment videos.

5 Conclusion and Limitations

We present a parkour learning system for low-cost robots. We propose a two-stage reinforcement learning method for overcoming hard exploration problems for learning parkour skills. We also extensive test our system in both simulation and the real world and show that our system has robust performance in various solving challenging parkour skills in challenging indoor and outdoor environments. However, the current system requires the simulation environments to be manually constructed. As a result, new skills can only be learned when new environments with different obstacles and appearances are added to the simulation. This poses a foundational limitation on the efficiency of developing agile skills. In the future, we hope to leverage recent advances in 3D vision and graphics to construct diverse simulation environments automatically from large-scale real-world data. And possibly we can train a parkour policy directly from RGB instead of depth.

References

[1] Merriam webster: Parkour. URL https://www.merriam-webster.com/dictionary/parkour.

[2] A. E. Patla. Understanding the roles of vision in the control of human locomotion. *Gait & posture*, 1997.

[3] A. A. Mohagheghi, R. Moraes, and A. E. Patla. The effects of distant and on-line visual information on the control of approach phase and step over an obstacle during locomotion. *Experimental brain research*, 2004.

[4] J. M. Loomis, J. A. Da Silva, N. Fujita, and S. S. Fukusima. Visual space perception and visually directed action. *Journal of experimental psychology: Human Perception and Performance*, 1992.

[5] J. S. Matthis and B. R. Fajen. Visual control of foot placement when walking over complex terrain. *Journal of experimental psychology: human perception and performance*, 2014.

[6] D. L. Puddle and P. S. Maulder. Ground reaction forces and loading rates associated with parkour and traditional drop landing techniques. *Journal of sports science & medicine*, 2013.

[7] Boston dynamics: Atlas. URL https://www.bostondynamics.com/atlas.

[8] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. In *RSS*, 2021.

[9] 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 *RSS*, 2018.

[10] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, Oct. 2020.

[11] 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*, 2019.

[12] Z. Xie, X. Da, M. van de Panne, B. Babich, and A. Garg. Dynamics randomization revisited: A case study for quadrupedal locomotion. In *ICRA*, 2021.

[13] X. Song, Y. Yang, K. Choromanski, K. Caluwaerts, W. Gao, C. Finn, and J. Tan. Rapidly adaptable legged robots via evolutionary meta-learning. In *IROS*, 2020.

[14] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine. Learning to walk via deep reinforcement learning. *arXiv preprint arXiv:1812.11103*, 2018.

[15] L. Smith, J. C. Kew, X. B. Peng, S. Ha, J. Tan, and S. Levine. Legged robots that keep on learning: Fine-tuning locomotion policies in the real world. In *ICRA*, 2022.

[16] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid locomotion via reinforcement learning. *RSS*, 2022.

[17] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang. Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers. In *ICLR*, 2022.

[18] W. Yu, V. C. V. Kumar, G. Turk, and C. K. Liu. Sim-to-real transfer for biped locomotion. In *IROS*, 2019.

[19] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots. In *ICRA*, 2021.

[20] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning. *arXiv preprint arXiv:2105.08328*, 2021.

[21] S. Ha, P. Xu, Z. Tan, S. Levine, and J. Tan. Learning to walk in the real world with minimal human effort. *arXiv preprint arXiv:2002.08550*, 2020.

[22] X. Da, Z. Xie, D. Hoeller, B. Boots, A. Anandkumar, Y. Zhu, B. Babich, and A. Garg. Learning a contact-adaptive controller for robust, efficient legged locomotion. *arXiv preprint arXiv:2009.10019*, 2020.

[23] Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, and D. Pathak. Coupling vision and proprioception for navigation of legged robots. In *CVPR*, 2022.

[24] S. Kareer, N. Yokoyama, D. Batra, S. Ha, and J. Truong. Vinl: Visual navigation and locomotion over obstacles. *ICRA*, 2023.

[25] M. Seo, R. Gupta, Y. Zhu, A. Skoutnev, L. Sentis, and Y. Zhu. Learning to walk by steering: Perceptive quadrupedal locomotion in dynamic environments. *ICRA*, 2023.

[26] J. Truong, A. Zitkovich, S. Chernova, D. Batra, T. Zhang, J. Tan, and W. Yu. Indoorsim-to-outdoorreal: Learning to navigate outdoors without any outdoor experience. *arXiv preprint arXiv:2305.01098*, 2023.

[27] 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*, Jan. 2022.

[28] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on Robot Learning (CoRL)*, 2022.

[29] R. Yang, G. Yang, and X. Wang. Neural volumetric memory for visual locomotion control. *CVPR*, 2023.

[30] W. Yu, D. Jain, A. Escontrela, A. Iscen, P. Xu, E. Coumans, S. Ha, J. Tan, and T. Zhang. Visual-locomotion: Learning to walk on complex terrains with vision. In *5th Annual Conference on Robot Learning*, 2021.

[31] A. Loquercio, A. Kumar, and J. Malik. Learning visual locomotion with cross-modal supervision. *arXiv preprint arXiv:2211.03785*, 2022.

[32] D. Hoeller, N. Rudin, C. Choy, A. Anandkumar, and M. Hutter. Neural scene representation for locomotion on structured terrain. *IEEE Robotics and Automation Letters*, 2022.

[33] X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, and S. Levine. Learning agile robotic locomotion skills by imitating animals. In *RSS*, 2020.

[34] Z. Fu, A. Kumar, J. Malik, and D. Pathak. Minimizing energy consumption leads to the emergence of gaits in legged robots. In *CoRL*, 2021.

[35] T. Li, J. Won, S. Ha, and A. Rai. Model-based motion imitation for agile, diverse and generalizable quadupedal locomotion. *arXiv preprint arXiv:2109.13362*, 2021.

[36] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots. Fast and efficient locomotion via learned gait transitions. In *CoRL*, 2021.

[37] C. Yang, K. Yuan, Q. Zhu, W. Yu, and Z. Li. Multi-expert learning of adaptive legged locomotion. *Science Robotics*, 2020.

[38] D. Kang, J. Cheng, M. Zamora, F. Zargarbashi, and S. Coros. Rl+ model-based control: Using on-demand optimal control to learn versatile legged locomotion. *arXiv preprint arXiv:2305.17842*, 2023.

[39] Z. Fu, X. Cheng, and D. Pathak. Deep whole-body control: learning a unified policy for manipulation and locomotion. In *Conference on Robot Learning*, 2022.

[40] Y. Ma, F. Farshidian, T. Miki, J. Lee, and M. Hutter. Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators. *IEEE Robotics and Automation Letters*, 2022.

[41] L. Smith, J. C. Kew, T. Li, L. Luu, X. B. Peng, S. Ha, J. Tan, and S. Levine. Learning and adapting agile locomotion skills by transferring experience. *RSS*, 2023.

[42] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011.

[43] H.-W. Park, P. M. Wensing, S. Kim, et al. Online planning for autonomous running jumps over obstacles in high-speed quadrupeds. *RSS*, 2015.

[44] Q. Nguyen, M. J. Powell, B. Katz, J. Di Carlo, and S. Kim. Optimized jumping on the mit cheetah 3 robot. In *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[45] C. Nguyen, L. Bao, and Q. Nguyen. Continuous jumping for legged robots on stepping stones via trajectory optimization and model predictive control. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 93–99. IEEE, 2022.

[46] C. Gehring, S. Coros, M. Hutter, C. D. Bellicoso, H. Heijnen, R. Diethelm, M. Bloesch, P. Fankhauser, J. Hwangbo, M. Hoepflinger, et al. Practice makes perfect: An optimization-based approach to controlling agile motions for a quadruped robot. *IEEE Robotics & Automation Magazine*, 2016.

[47] C. Wright, A. Buchan, B. Brown, J. Geist, M. Schwerin, D. Rollinson, M. Tesch, and H. Choset. Design and architecture of the unified modular snake robot. In *2012 IEEE international conference on robotics and automation*, 2012.

[48] A. M. Johnson, T. Libby, E. Chang-Siu, M. Tomizuka, R. J. Full, and D. E. Koditschek. Tail assisted dynamic self righting. In *Adaptive mobile robotics*. 2012.

[49] C. D. Bellicoso, K. Krämer, M. Stäuble, D. Sako, F. Jenelten, M. Bjelonic, and M. Hutter. Alma-articulated locomotion and manipulation for a torque-controllable robot. In *2019 International conference on robotics and automation (ICRA)*, pages 8477–8483. IEEE, 2019.

[50] Q. Nguyen, A. Agrawal, X. Da, W. C. Martin, H. Geyer, J. W. Grizzle, and K. Sreenath. Dynamic walking on randomly-varying discrete terrain with one-step preview. In *Robotics: Science and Systems*, 2017.

[51] R. Antonova, A. Rai, and C. G. Atkeson. Deep kernels for optimizing locomotion controllers. In *Conference on Robot Learning*, 2017.

[52] M. H. Raibert, H. B. Brown Jr, and M. Chepponis. Experiments in balance with a 3d one-legged hopping machine. *The International Journal of Robotics Research*, 1984.

[53] Cassie sets a guinness world record. URL https://agilityrobotics.com/news/2022/cassie-sets-a-guinness-world-record.

[54] 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*, 2022.

[55] Y. Ma, F. Farshidian, and M. Hutter. Learning arm-assisted fall damage reduction and recovery for legged mobile manipulators. *arXiv preprint arXiv:2303.05486*, 2023.

[56] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath. Robust and versatile bipedal jumping control through multi-task reinforcement learning. *arXiv preprint arXiv:2302.09450*, 2023.

[57] Y. Yang, X. Meng, W. Yu, T. Zhang, J. Tan, and B. Boots. Continuous versatile jumping using learned action residuals. *L4DC*, 2023.

[58] G. B. Margolis, T. Chen, K. Paigwar, X. Fu, D. Kim, S. Kim, and P. Agrawal. Learning to jump from pixels. *CoRL*, 2021.

[59] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter. Advanced skills by learning locomotion and local navigation end-to-end. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[60] C. Li, M. Vlastelica, S. Blaes, J. Frey, F. Grimminger, and G. Martius. Learning agile skills via adversarial imitation of rough partial demonstrations. In *Conference on Robot Learning*, 2023.

[61] X. Cheng, A. Kumar, and D. Pathak. Legs as manipulator: Pushing quadrupedal agility beyond locomotion. *ICRA*, 2023.

[62] E. Arcari, M. V. Minniti, A. Scampicchio, A. Carron, F. Farshidian, M. Hutter, and M. N. Zeilinger. Bayesian multi-task learning mpc for robotic mobile manipulation. *IEEE Robotics and Automation Letters*, 2023.

[63] H. Ito, K. Yamamoto, H. Mori, and T. Ogata. Efficient multitask learning with an embodied predictive model for door opening and entry with whole-body control. *Science Robotics*, 2022.

[64] A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *ICLR*, 2019.

[65] B. Forrai, T. Miki, D. Gehrig, M. Hutter, and D. Scaramuzza. Event-based agile object catching with a quadrupedal robot. *ICRA*, 2023.

[66] H. Huang, A. Loquercio, A. Kumar, N. Thakkar, K. Goldberg, and J. Malik. More than an arm: Using a manipulator as a tail for enhanced stability in legged locomotion. *arXiv preprint arXiv:2305.01648*, 2023.

[67] T. Haarnoja, B. Moran, G. Lever, S. H. Huang, D. Tirumala, M. Wulfmeier, J. Humplik, S. Tunyasuvunakool, N. Y. Siegel, R. Hafner, et al. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *arXiv preprint arXiv:2304.13653*, 2023.

[68] Y. Ji, G. B. Margolis, and P. Agrawal. Dribblebot: Dynamic legged manipulation in the wild. *arXiv preprint arXiv:2304.01159*, 2023.

[69] X. Huang, Z. Li, Y. Xiang, Y. Ni, Y. Chi, Y. Li, L. Yang, X. B. Peng, and K. Sreenath. Creating a dynamic quadrupedal robotic goalkeeper with reinforcement learning. *arXiv preprint arXiv:2210.04435*, 2022.

[70] Y. Ji, Z. Li, Y. Sun, X. B. Peng, S. Levine, G. Berseth, and K. Sreenath. Hierarchical reinforcement learning for precise soccer shooting skills using a quadrupedal robot. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[71] K. Caluwaerts, A. Iscen, J. C. Kew, W. Yu, T. Zhang, D. Freeman, K.-H. Lee, L. Lee, S. Saliceti, V. Zhuang, et al. Barkour: Benchmarking animal-level agility with quadruped robots. *arXiv preprint arXiv:2305.14654*, 2023.

[72] P. Fankhauser, M. Bloesch, and M. Hutter. Probabilistic terrain mapping for mobile robots with uncertain localization. *IEEE Robotics and Automation Letters*, 2018.

[73] I.-S. Kweon, M. Hebert, E. Krotkov, and T. Kanade. Terrain mapping for a roving planetary explorer. In *IEEE International Conference on Robotics and Automation*, 1989.

[74] I.-S. Kweon and T. Kanade. High-resolution terrain map from multiple sensor data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992.

[75] A. Kleiner and C. Dornhege. Real-time localization and elevation mapping within urban search and rescue scenarios. *Journal of Field Robotics*, 2007.

[76] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis. Real-time trajectory adaptation for quadrupedal locomotion using deep reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[77] D. Belter, P. Łabcki, and P. Skrzypczyński. Estimating terrain elevation maps from sparse and uncertain multi-sensor data. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2012.

[78] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart. Robot-centric elevation mapping with uncertainty estimates. In *Mobile Service Robotics*. 2014.

[79] Y. Pan, X. Xu, Y. Wang, X. Ding, and R. Xiong. Gpu accelerated real-time traversability mapping. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2019.

[80] B. Yang, L. Wellhausen, T. Miki, M. Liu, and M. Hutter. Real-time optimal navigation planning using learned motion costs. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[81] R. O. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti. Learning ground traversability from simulations. *IEEE Robotics and Automation letters*, 2018.

[82] J. Guzzi, R. O. Chavez-Garcia, M. Nava, L. M. Gambardella, and A. Giusti. Path planning with local motion estimations. *IEEE Robotics and Automation Letters*, 2020.

[83] S. Yang, Z. Zhang, Z. Fu, and Z. Manchester. Cerberus: Low-drift visual-inertial-leg odometry for agile locomotion. *ICRA*, 2023.

[84] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart. Robust visual inertial odometry using a direct ekf-based approach. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 298–304. IEEE, 2015.

[85] D. Wisth, M. Camurri, and M. Fallon. Vilens: Visual, inertial, lidar, and leg odometry for all-terrain legged robots. *IEEE Transactions on Robotics*, 2022.

[86] D. Wisth, M. Camurri, S. Das, and M. Fallon. Unified multi-modal landmark tracking for tightly coupled lidar-visual-inertial odometry. *IEEE Robotics and Automation Letters*, 6(2): 1004–1011, 2021.

[87] R. Buchanan, M. Camurri, F. Dellaert, and M. Fallon. Learning inertial odometry for dynamic legged robot state estimation. In *Conference on Robot Learning*, pages 1575–1584. PMLR, 2022.

[88] S. Yang, H. Kumar, Z. Gu, X. Zhang, M. Travers, and H. Choset. State estimation for legged robots using contact-centric leg odometry. *arXiv preprint arXiv:1911.05176*, 2019.

[89] D. Wisth, M. Camurri, and M. Fallon. Robust legged robot state estimation using factor graph optimization. *IEEE Robotics and Automation Letters*, 2019.

[90] S. Omari, M. Bloesch, P. Gohl, and R. Siegwart. Dense visual-inertial navigation system for mobile robots. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2634–2640. IEEE, 2015.

[91] C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, 2014.

[92] D. Scaramuzza and F. Fraundorfer. Visual odometry [tutorial]. *IEEE robotics & automation magazine*, 2011.

[93] M. DeDonato, V. Dimitrov, R. Du, R. Giovacchini, K. Knoedler, X. Long, F. Polido, M. A. Gennert, T. Padır, S. Feng, et al. Human-in-the-loop control of a humanoid robot for disaster response: a report from the darpa robotics challenge trials. *Journal of Field Robotics*, 2015.

[94] F. Jenelten, T. Miki, A. E. Vijayan, M. Bjelonic, and M. Hutter. Perceptive locomotion in rough terrain–online foothold optimization. *IEEE Robotics and Automation Letters*, 2020.

[95] D. Kim, D. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, and S. Kim. Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[96] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter. Navigation planning for legged robots in challenging terrain. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[97] A. Chilian and H. Hirschmüller. Stereo camera based navigation of mobile robots on rough terrain. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[98] C. Mastalli, I. Havoutis, A. W. Winkler, D. G. Caldwell, and C. Semini. On-line and on-board planning and perception for quadrupedal locomotion. In *2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, 2015.

[99] A. Agrawal, S. Chen, A. Rai, and K. Sreenath. Vision-aided dynamic quadrupedal locomotion on discrete terrain using motion libraries. In *2022 International Conference on Robotics and Automation (ICRA)*, 2022.

[100] J. Z. Kolter, M. P. Rodgers, and A. Y. Ng. A control architecture for quadruped locomotion over rough terrain. In *2008 IEEE International Conference on Robotics and Automation*, 2008.

[101] M. Kalakrishnan, J. Buchli, P. Pastor, and S. Schaal. Learning locomotion over rough terrain using terrain templates. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[102] L. Wellhausen and M. Hutter. Rough terrain navigation for legged robots using reachability planning and template learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[103] C. Mastalli, M. Focchi, I. Havoutis, A. Radulescu, S. Calinon, J. Buchli, D. G. Caldwell, and C. Semini. Trajectory and foothold optimization using low-dimensional models for rough terrain locomotion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[104] O. A. V. Magana, V. Barasuol, M. Camurri, L. Franceschi, M. Focchi, M. Pontil, D. G. Caldwell, and C. Semini. Fast and continuous foothold adaptation for dynamic locomotion through cnns. *IEEE Robotics and Automation Letters*, 2019.

[105] F. L. G. Bermudez, R. C. Julian, D. W. Haldane, P. Abbeel, and R. S. Fearing. Performance analysis and terrain classification for a legged robot over rough terrain. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 513–519. IEEE, 2012.

[106] V. Tsounis, M. Alge, J. Lee, F. Farshidian, and M. Hutter. Deepgait: Planning and control of quadrupedal gaits using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 2020.

[107] X. B. Peng, G. Berseth, K. Yin, and M. Van De Panne. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 2017.

[108] 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*, 2022.

[109] D. Jain, A. Iscen, and K. Caluwaerts. From pixels to legs: Hierarchical learning of quadruped locomotion. *arXiv preprint arXiv:2011.11722*, 2020.

[110] A. Escontrela, G. Yu, P. Xu, A. Iscen, and J. Tan. Zero-shot terrain generalization for visual locomotion policies. *arXiv preprint arXiv:2011.05513*, 2020.

[111] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[112] R. Tedrake. Trajectory optimization. *Underactuated Robotics*. URL http://underactuated.mit.edu/trajopt.html.

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

[114] 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.

[115] Y. Burda, H. Edwards, A. Storkey, and O. Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.