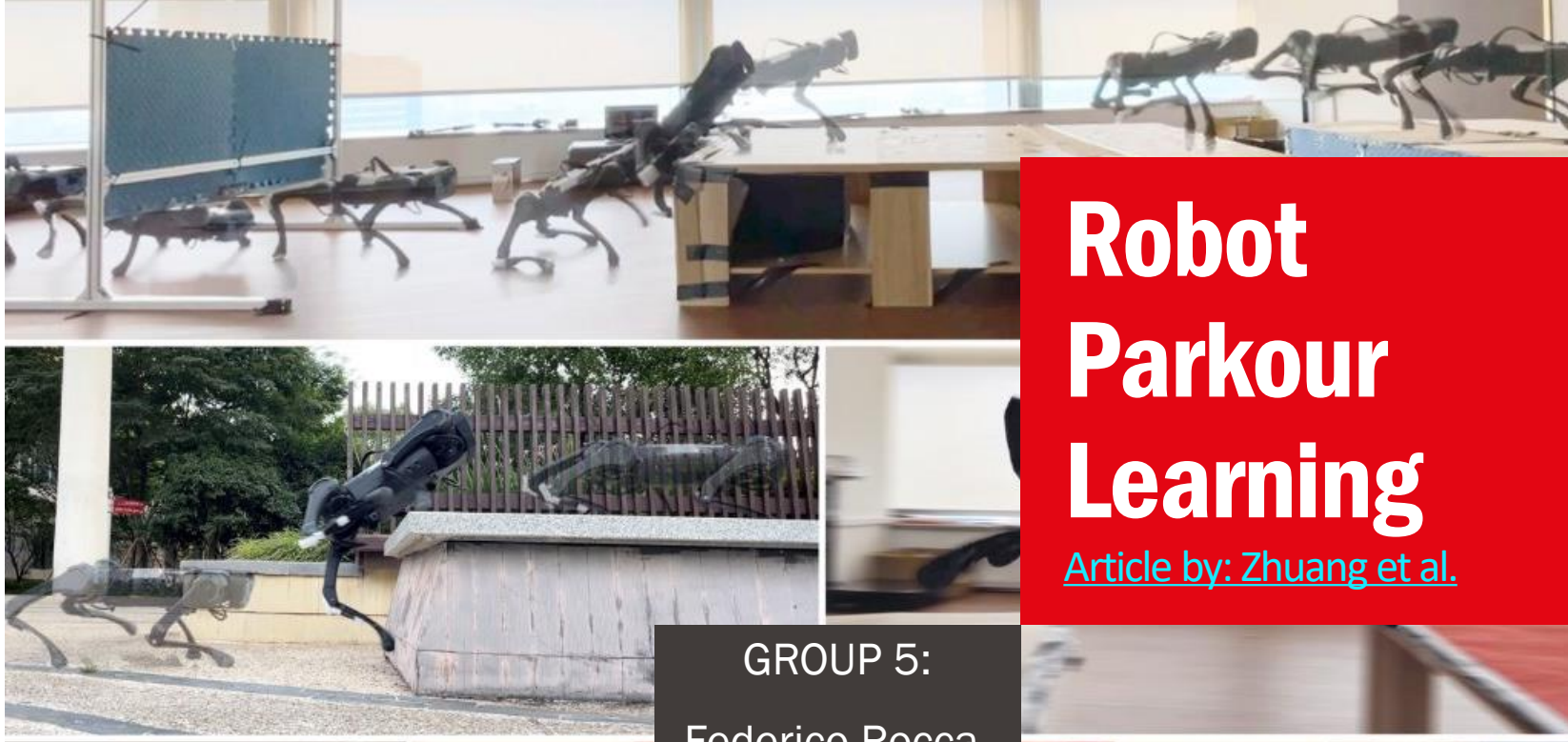
EPFL



Robot Parkour Learning

Article by: Zhuang et al.

GROUP 5:

Federico Rocca,
Niel Mistry &
Yugo Kadowaki

Main Ideas

- Objective
  - Development of learning methods for getting **parkour abilities in low-cost quadrupedal robots**
- Methodology
  - Extensive experiments in simulation and the real world
  - Generalization to different robots
- Contribution
  - Open-source system for robot parkour learning
  - **Two-stage RL method,** uses **simple reward function**

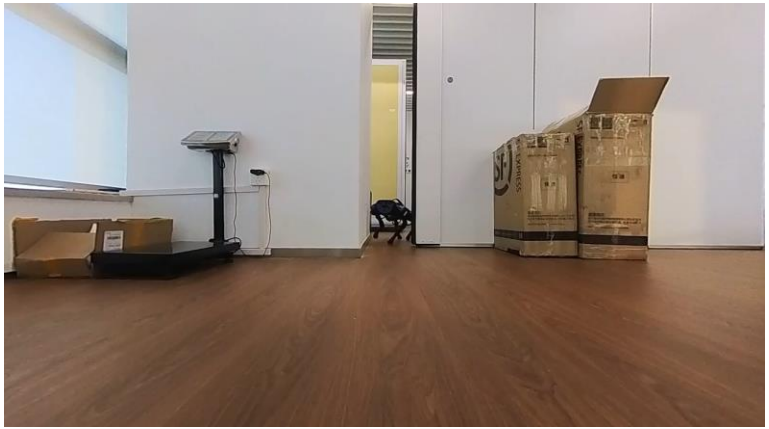
Go1, Unitree Robotics

A1, Unitree Robotics

Climb

Leap

Crawl



Tilt

Executive Summary of Key Aspects

- Robot: **low-cost** quadrupedal robot (Unitree A1 and Go1)
- Control:
  - **Parkour policy** (50Hz) with **GRU** for Temporal Reasoning and **MLP** for Action Generation
  - **Vision based control** (10Hz) with **CNN** for Depth Embedding
  - **Joint PD Control** (1000Hz) with torque limiting
- Design method: **two-stage** reinforcement learning (pre-training and fine tuning) + distillation
- Gait: **parkour skills**
  - Running (default gait)
  - Climbing
  - Leaping
  - Crawling
  - Squeezing through (tilting)
- Sensors:
  - **Depth camera** (Intel RealSense D435 depth camera)
  - **Proprioception**:
    - Joint Positions (at each of the 12 joints in the robot)
    - Joint Velocities
    - Base Orientation (roll, pitch and yaw)
    - Base Angular Velocities

Stages of Policy Development

1. Virtual RL Pre-Training
   - Soft Constraints: Simulated robot can **go through** obstacles
   - Negative Reward for going through obstacles
   - Training gets harder progressively by larger gaps, higher cliffs, etc.
2. Virtual RL Fine-Tuning
   - Hard Constraints: Robot can **no longer go through** obstacles
   - Fine-tune the behaviors learned in the pre-training stage with realistic dynamics
3. Distillation of final policy
   - Expert demonstration from the specialized policy
   - Final policy doesn’t need privileged information

Soft Constraints

- Challenging learning environment ⇛ generic RL algorithms not effective
- Pre-training with *soft* constraints: robot is permitted to violate, but with penalty
  - Keeps training going even when local minima cause collisions
  - Negative reward gradually enforces constraints with automatic curriculum
- Reward:

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

- p: collision points, $d(p)$:penetration depth
  $\mathbb{1}[p]$: indicator function
- Multiplied by v_x to prevent cheating by sprinting through obstacle

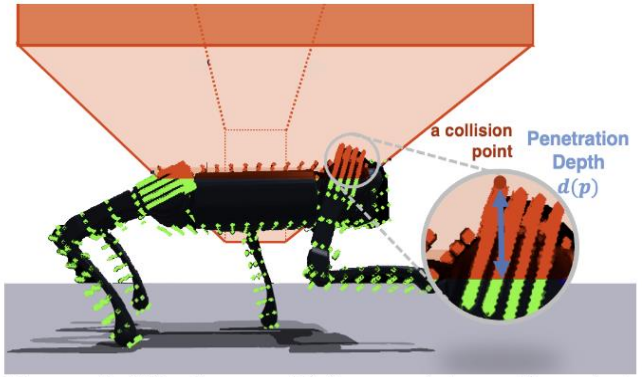
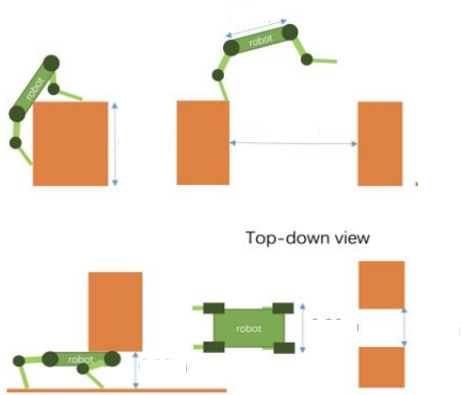


Figure 4: We show collisions points on the robot. Collision points that penetrate obstacles are in red.

Automatic Curriculum

- Obstacles start easy, difficulty grows as scores improve
- Threshold on penetration reward used to modulate difficulty:
  - Reward higher than a threshold: difficulty score s raised by a unit
  - Reward lower than a threshold: difficulty score s lowered by a unit
- Obstacle property set by $(1-s)l_{easy} + sl_{hard}$

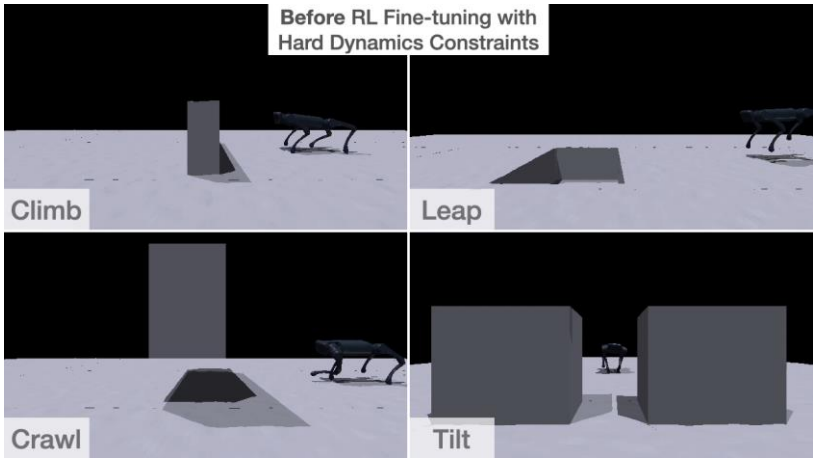


Skill	Obstacle Properties	Training Ranges ($[l_{\text{easy}}, l_{\text{hard}}]$)	Test Ranges ($[l_{\text{easy}}, l_{\text{hard}}]$)
Climb	obstacle height	[0.2, 0.45]	[0.25, 0.5]
Leap	gap length	[0.2, 0.8]	[0.3, 0.9]
Crawl	clearance	[0.32, 0.22]	[0.3, 0.2]
Tilt	path width	[0.32, 0.28]	[0.3, 0.26]

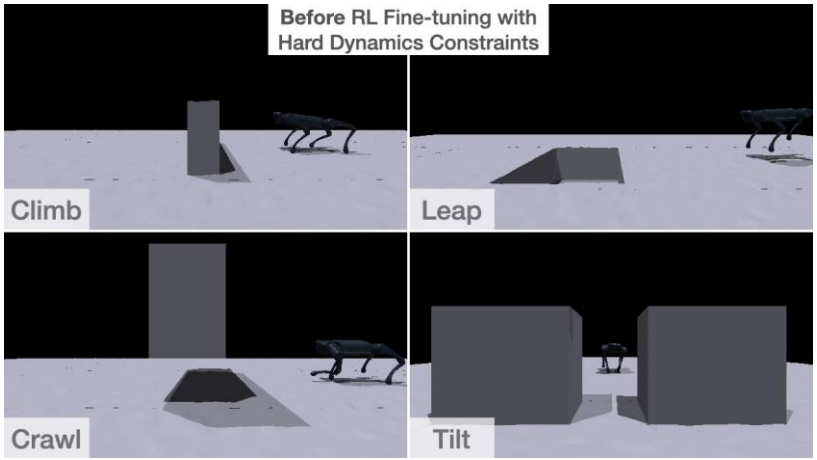
Table 1: Ranges for obstacle properties for each skill during training, measured in meters.

RL Fine-Tuning

- Fine-Tuning
  - **Enforce all dynamics constraints and fine-tune the behaviors learned in the pre-training stage with realistic dynamics**
- Conditions
  - No penetrations are allowed in the Fine-Tuning process
  - Choose obstacle properties(height, gap, clearance, gap width) randomly for each skill



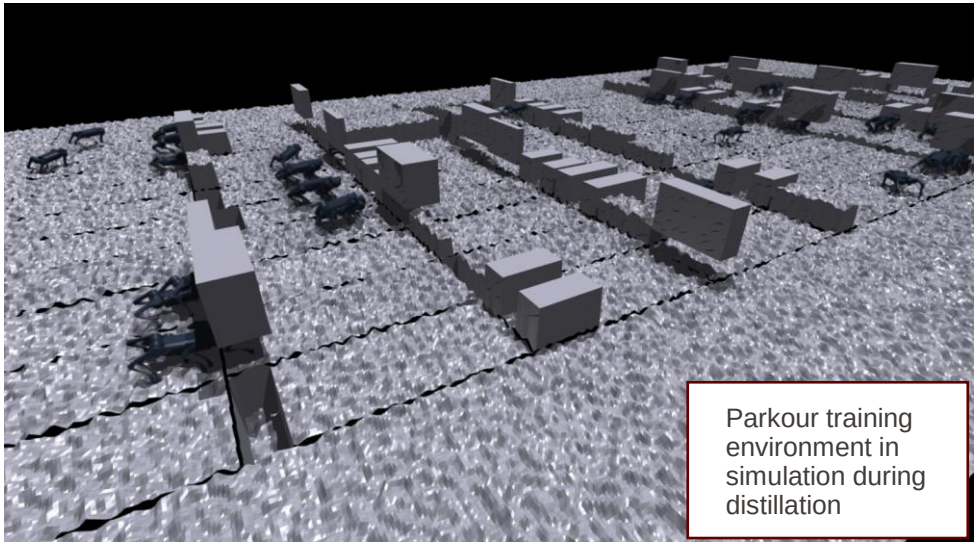
Before RL Fine-tunning



After RL Fine-tunning

Distillation

- Merges the 5 specialized parkour policies into a single vision-based policy
  - Doesn't use privileged visual (e_t^{vis}) and physics (e_t^{phy}) information, contrary to the specialized policies
- Implemented using DAgger (Dataset Aggregation)
  - Trains the single policy by iteratively improving it using expert demonstration from the specialized policies
- Samples obstacles types and properties and assigns the corresponding specialized skill policy to teach the parkour policy how to act at a state



Parkour training environment in simulation during distillation

Distilling the Policy

- Policy parametrized as GRU (Gated Recurrent Unit), with input:
  - Recurrent latent state
  - Proprioception $s_t^{proprio}$
  - Previous action a_{t-1}
  - Latent embedding of depth image I_t^{depth}, processed by a CNN
- Distillation objective:

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

- $\theta_{parkour}$: network parameters of the parkour policy
- sim: simulator with hard dynamics constraints
- D: divergence function, binary cross entropy loss

Sim-to-Real and Deployment

- Distillation can bridge the sim-to-real gap in physical dynamics properties
- Overcome differences between the rendered depth image in simulation and the onboard depth image in the real world
  - Apply depth-clipping, pixel-level Gaussian noise and random artifacts to rendered depth images
  - Apply depth-clipping, hole-filling and spatial and temporal smoothing to real world depth images
- Refresh rate of images is 10Hz, parkour policy at 50Hz uses the latest embedding of the depth image processed by the CNN
- Output actions of the policy are target joint positions converted to torques at 1000Hz through PD controller (K_p=50, K_d=1)
- Torque limits of 25Nm for safe deploying

Results, Simulation experiments

- Blind vs Ours →Visual information is crucial for learning parkour behavior
- w/o Soft Dyn vs w/ Soft Dyn(Ours) → Pre-training with soft dynamics enables robots to learn every parkour skills with higher success rate (around 95%)
- No Distill vs Ours → Distillation is effective for Sim2Real

	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 3 random seeds. Our parkour policy shows the best performance using only sensors that are available in the real world. We evaluate on the test environments with obstacles proprieties that are more difficult than the ones of training environments shown in Table 1.

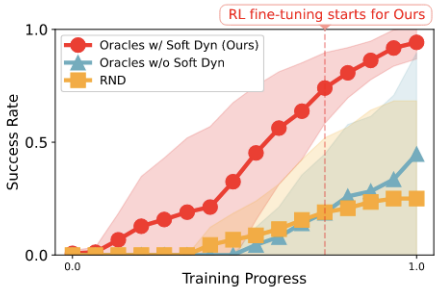


Figure 7: Comparison of specialized oracles trained with soft dynamics constraints with baselines averaged across every skill and three trials.

Results, Real-world experiments

- Proposed policy achieves the best performance in any parkour skills (Climb, Leap, Crawl, and Tilt)

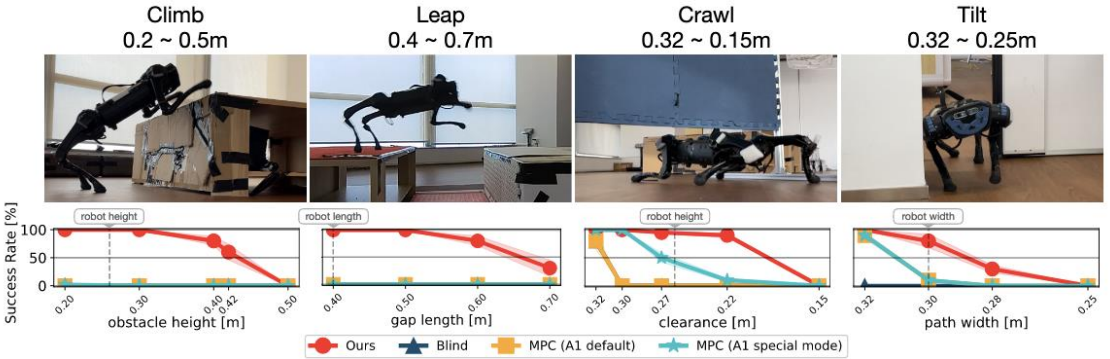


Figure 6: Real-world indoor quantitative experiments. Our parkour policy can achieve the best performance, compared with a blind policy and built-in MPC controllers. We control the MPC in A1 special mode by teleoperating the robot lower down or tilt the body during crawling and tilt respectively.

- Emergent Re-trying Behaviors

- Failure cases
  - Failed to climb very high obstacles of 0.8m(Three times of robot's height)
  - Failed to leap on low-friction platform
  - Failed to climb a high and soft obstacle

External Citations

- Influences
  - *Zhuang, Ziwen et al. “Humanoid Parkour Learning.” ArXiv abs/2406.10759 (2024):* same group, uses similar strategy on humanoids
- Criticism
  - *Chane-Sane, Elliot et al. “SoloParkour: Constrained Reinforcement Learning for Visual Locomotion from Privileged Experience.” ArXiv abs/2409.13678 (2024):* assert that assumption of privileged information being reconstructed from history of depth images is unrealistic
  - *Luo, Shixin et al. “PIE: Parkour With Implicit-Explicit Learning Framework for Legged Robots.” IEEE Robotics and Automation Letters 9 (2024):* similar criticism, privileged information associated with geometric properties. If terrain can’t be described by these, then robot cannot cope
  - *Cheng, Xuxin et al. “Extreme Parkour with Legged Robots.” 2024 IEEE International Conference on Robotics and Automation (ICRA) (2023):* Geometrical privileged information does not generalize, they use scandots instead. Also, they criticize the complexity of the curriculum

Pros and Cons

Pros

- Easy-to-understand reward functions
- Emergence of re-trying behaviours
- Ability to use on low-cost robots with raw depth images and proprioception
- Generalization to different robots

Cons

- New skills cannot be learned easily because the simulation environment needs to be manually constructed
- Not generalizable to terrain that cannot be described by the privileged information (rough surfaces, unmodelled features)

Possible Exam Questions

- What are the advantages of soft constraints used in reinforcement learning pre-training of curricula with obstacles?
  - Soft constraints allow a larger exploration space during the early stages of the training process when the controller has a high chance of collision.
  - The training episode can continue even after a collision with a soft constraint, which allows that episode to be useful
- What important aspect of the sim-to-real transfer is overcome by the distillation phase?
  - The trained specialized policies require privileged physics and visual information as inputs, in order to select the best action, but that information is not available in the real world. The distillation generates a single parkour policy that runs by only accessing proprioception information and visual information from the depth camera.

Thank You!

Questions?