EPFL

Article presentation: Learning stable bipedal locomotion skills for quadrupedal robots on challenging terrains with automatic fall recovery

(Xiao, E., Dong, Y., Lam, J. et al. npj Robot 3, 22, 2025, https://doi.org/10.1038/s44182-025-00043-2)

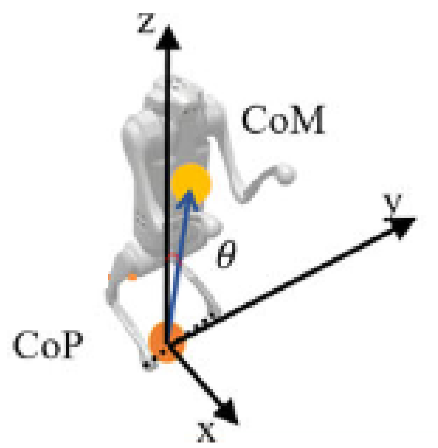
Jules Raphael Simon Streit
Fatih Selim Yilmaz
Cyril Nils Goffin

Group 16


is extremely challenging,
Sand beach (high friction)

What is it about?

- Quadruped robot learns **stable bipedal walking** with **reinforcement learning**
- Presents a new controller to keep **balance via CoM–CoP estimation**
- Trained **in simulation** using only **proprioceptive sensors** (no vision)
- Enables **quad⟷biped transitions** and **fall recovery**
- Demonstrates **robust walking on uneven and soft terrain**
- Generalizes to **other quadruped morphologies**



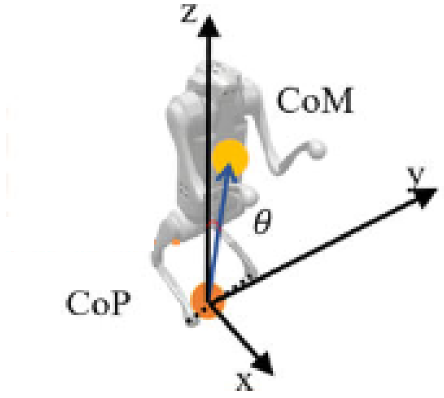
Key aspects

- **Robots:** quadrupeds (ANYmal, Mini-Cheetah, Go2, Lite3, Spot, and Alingo)
- **Gait:** bipedal walking, quad⟷biped transitions
- **Control:** torque control (PD + RL policy, 12 DoF)
- **Method:** PPO + EstimatorNet (CoM-CoP + velocity)
- **Sensors:** proprioception only (IMU + joints)
- **Training:** Isaac Gym, ~2 h on RTX 3060 (1024 parallel agents)

Methods

Deep Reinforcement Learning (PPO Training)

CoM–CoP: vector indicating balance stability (distance between body mass center and ground support point).



1. Two-Block Controller Architecture

EstimatorNet: predicts *CoM–CoP* and *linear velocity*

Policy Network: outputs joint position targets

2. Reward Structure

- **Balance**: CoM–CoP angle, handle length, pendulum angle acceleration
- **Locomotion**: track commanded velocity
- **Penalty**: front-leg contact in bipedal mode

Reward Structure: Balance, Locomotion, Penalty

DeepRL Algorithm: Proximal Policy Optimization (PPO)

3. DRL Algorithm PPO

Training Setup:

- Simulator: **Isaac Gym**
- **1024 parallel agents**
- **Training time: ~2 hours** (RTX 3060 Laptop GPU)
- Terrain: **flat only**

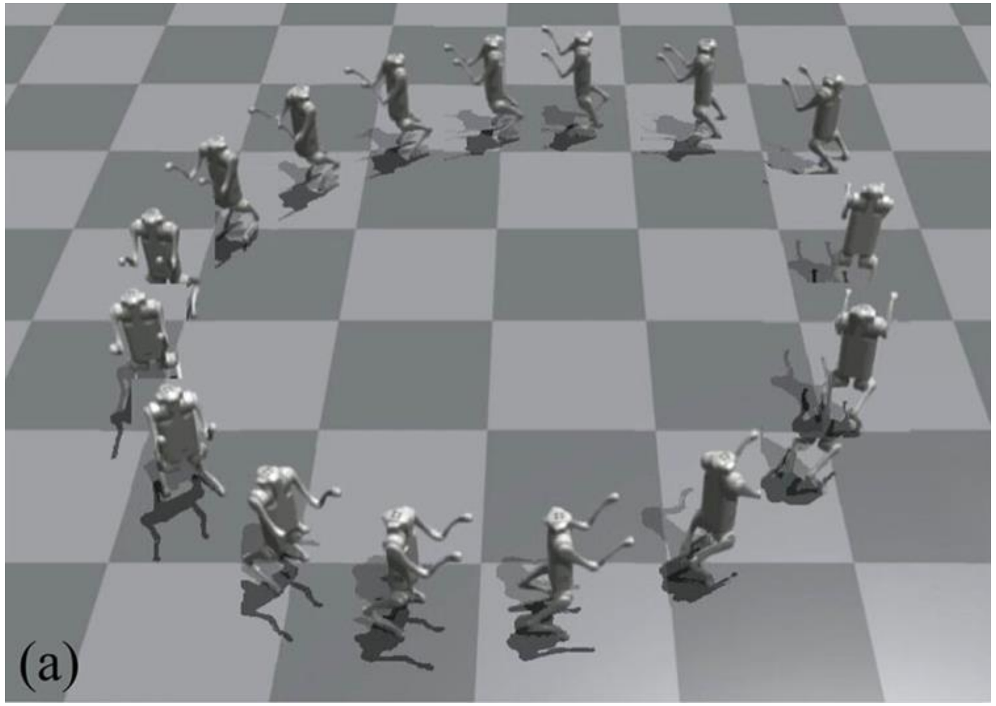
Methods

Deep Reinforcement Learning (PPO Training)

5. Fixed trained controller composed of:

Trained EstimatorNet:
predicts *CoM–CoP* and *linear velocity*

Trained Policy Network:
outputs joint position targets



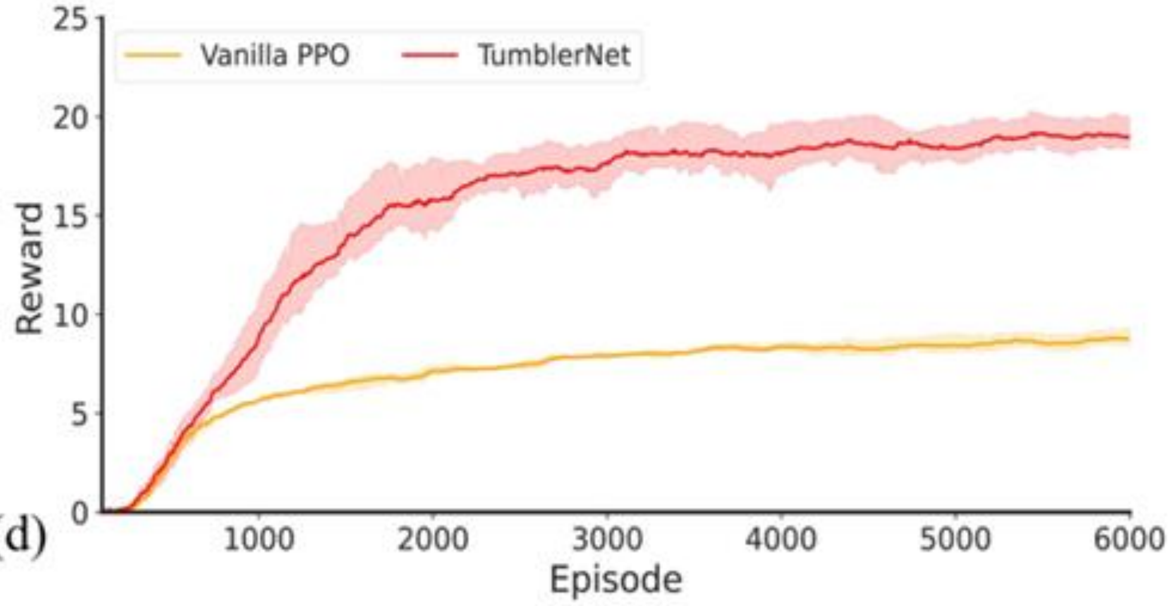
- **EstimatorNet** — predicts *CoM–CoP* and *linear velocity* from IMU + joints
- **Policy Network** — outputs joint position targets for locomotion

The trained controller runs in real time:

Sensors → EstimatorNet → Policy → Joint Commands (PD control)

Results

(d)

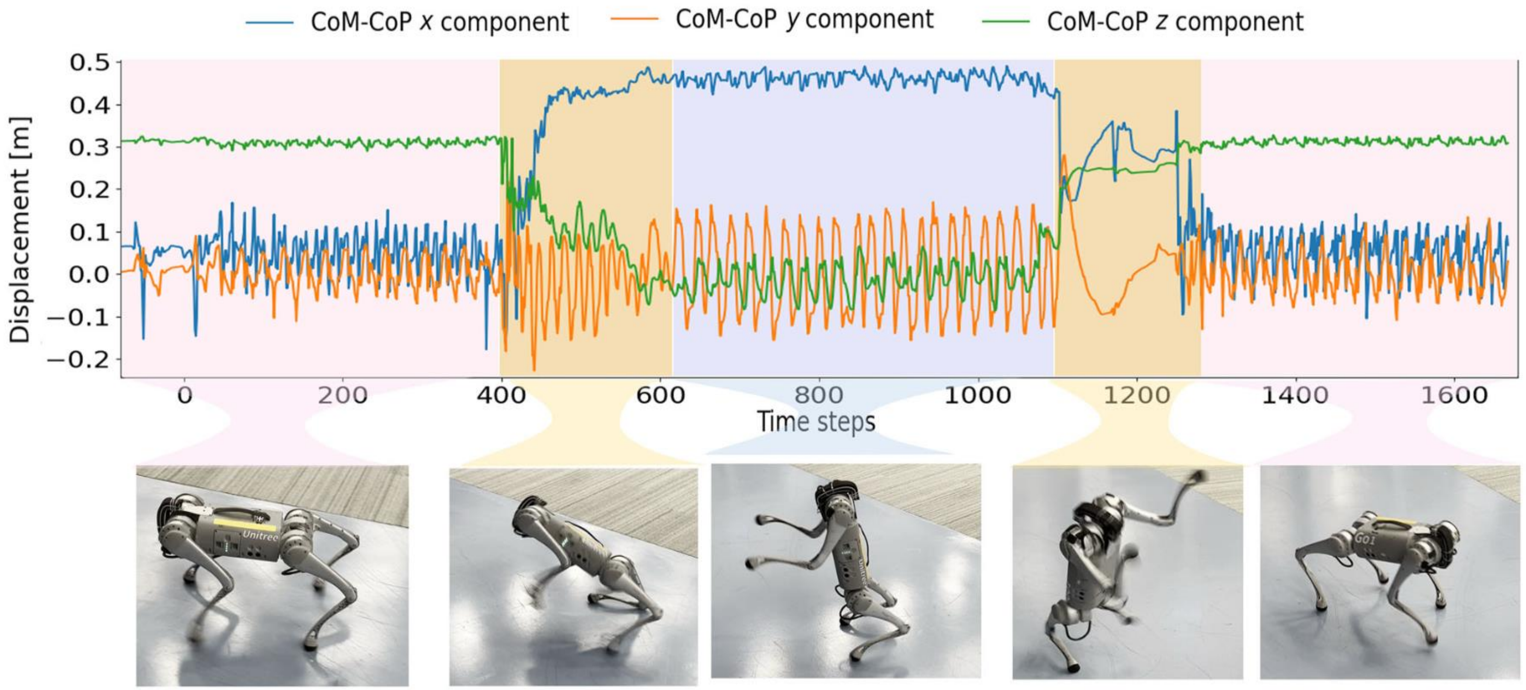


Bipedal locomotion with front and hind legs

- Training With/Without their estimator EstimatorNet: TumblerNet vs Vanilla proximal policy optimization (PPO)

Higher reward with TumblerNet

Results



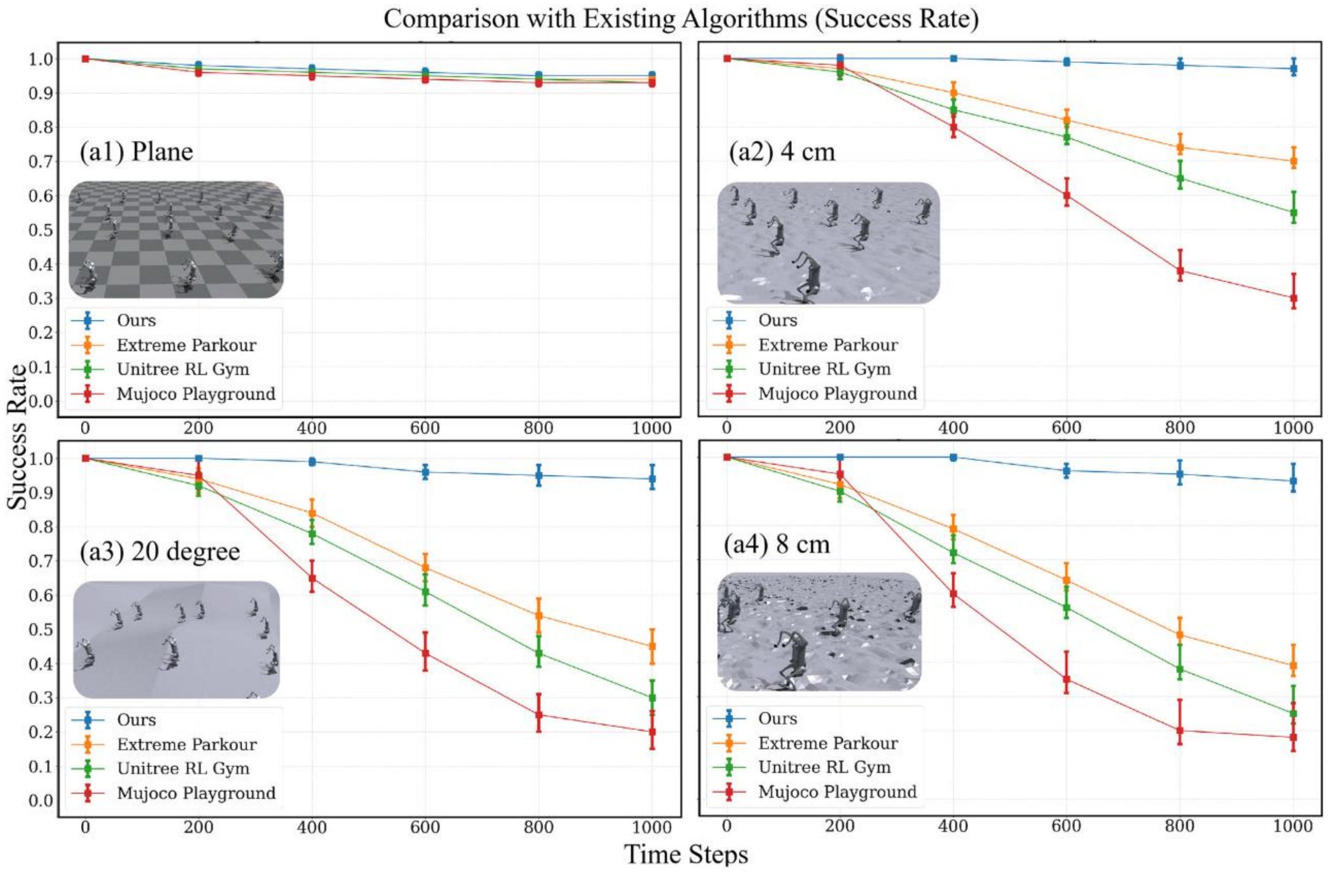
Transition : quadrupedal → bipedal → quadrupedal

Results

Robustness over challenging terrains:

- Indoor environments
- Uphill, softpad and downhill
- Uneven terrains made by planks
- Rocky field
- Sandy beach

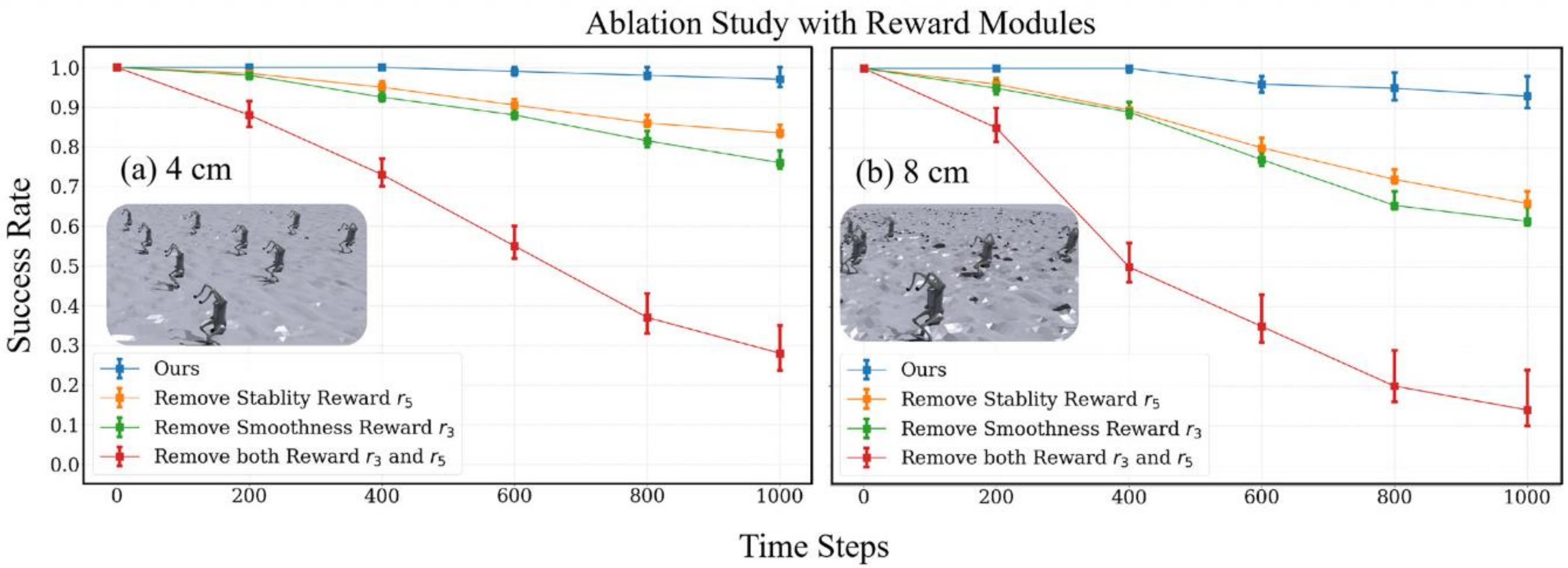
Results



Comparison with existing algorithms

- Flat plane
- 4 cm uneven trimesh terrain
- 20° slope terrain
- 8 cm uneven trimesh terrain

Results



5 reward criterias : tracking commands tasks **r1**, avoiding collision **r2**, smoothness of motion **r3**, bipedal locomotion **r4**, stability **r5**

Impact

MARG: MAstering Risky Gap Terrains for Legged Robots with Elevation Mapping
(Yinzhao Dong, Ji Ma, Liu Zhao, Wanyue Li)

- Published Aug 1, 2025 (Nature), **1 citation** (expected for a 3-month-old paper)
- Early interest on open-source repository, **24 stars and 3 forks** (community interest)
- No documented criticisms in the literature yet
- Early days, further uptake and citation are still pending

Pros & cons

- Strong generalization despite training **only on flat terrain**
- Unified controller for bipedal, quadrupedal, and **2↔4 leg transitions**
- Robust to external disturbances

- No perception, can’t plan footholds on complex terrain
- Limited long-term planning, **purely reactive controller**
- Strong dependence on accurate CoM–CoP estimation

Thank you for your attention!

Any questions?