ANIMAL ROBOTS

# Learning quadrupedal locomotion on deformable terrain

**Suyoung Choi, Gwanghyeon Ji, Jeongsoo Park, Hyeongjun Kim, Juhyeok Mun, Jeong Hyun Lee, Jemin Hwangbo***



**Simulation-based reinforcement learning approaches are leading the next innovations in legged robot control. However, the resulting control policies are still not applicable on soft and deformable terrains, especially at high speed. The primary reason is that reinforcement learning approaches, in general, are not effective beyond the data distribution: The agent cannot perform well in environments that it has not experienced. To this end, we introduce a versatile and computationally efficient granular media model for reinforcement learning. Our model can be parameterized to represent diverse types of terrain from very soft beach sand to hard asphalt. In addition, we introduce an adaptive control architecture that can implicitly identify the terrain properties as the robot feels the terrain. The identified parameters are then used to boost the locomotion performance of the legged robot. We applied our techniques to the Raibo robot, a dynamic quadrupedal robot developed in-house. The trained networks demonstrated high-speed locomotion capabilities on deformable terrains: The robot was able to run on soft beach sand at 3.03 meters per second although the feet were completely buried in the sand during the stance phase. We also demonstrate its ability to generalize to different terrains by presenting running experiments on vinyl tile flooring, athletic track, grass, and a soft air mattress.**

## INTRODUCTION

Recent studies on legged robots have predominantly focused on locomotion on rigid ground, and, consequently, the resulting controllers have compromising performance on soft and deformable terrains. Because a large portion of the terrestrial surface is covered by nonrigid substrates, such as sand, soil, and vegetation, locomotion on soft terrains is a necessary skill for many applications of legged robots. At the same time, traversing such terrain is more suited to legged locomotion as opposed to wheeled mobility, which often encounters sinkage and slippages (*1*). Because soft earthy substrates might yield even under a small load, legged robots should be able to adapt to different terrain characteristics as they feel the terrain, as animals do (*2*–*4*). Although recent advances in legged robotics have led to more versatile robots that can traverse a wider range of terrains (*5*–*8*), most of the existing works on control assumed that the terrain is rigid, and such an assumption inevitably leads to compromising locomotion performance on soft substrates (*9*).

There have been efforts to control legged robots on highly soft and deformable terrains. Some of them took advantage of known properties of the terrain (*10*–*12*) or estimated the ground reaction force with a nonparametric model trained with data from prior experiences (*13*). However, because the ground properties and prior experiences are often not available in the outdoor environments, demonstrations of those approaches have been confined to laboratory environments. The issue stems from the difficulty of predicting the terrain dynamics. In particular, terrain properties, such as compliance and deformability, vary with the weather and the surface conditions, and such differences are indistinguishable even to a highly advanced vision system. Furthermore, the complexity of a combined model that describes the interactions between the floating-based robot and the terrain leads to an impractically complex control architecture. As a result, contemporary research on integrating the terrain model into the legged robotic control has primarily been limited to morphologically simple robots, e.g., one-dimensional (1D) hopper (*10*, *12*, *13*).

Rising recently, the application of model-free reinforcement learning (RL) to legged robotics has demonstrated extensive generalization to unseen environments (*5*–*7*, *14*–*18*) using simulated environments. Specifically, quadrupedal locomotion controllers trained on rigid rough terrain exhibited robust locomotion capabilities on various challenging terrains, even including several deformable terrains such as mud and snow (*5*, *16*). One of the core ideas that led to such a success is the privileged learning (PL) framework. In this framework, a teacher policy is trained with privileged access to states only available in the simulation. A student policy learns to imitate the teacher's encoding of privileged information and act accordingly. Another key feature was the randomization of physical parameters during the training, which is often referred to as domain randomization (*19*). It can be combined with adaptation methods (*18*, *20*) such that the policy can be adaptive to a new environment from the training domain distribution. However, because it has a prior bias toward the trained environments, it fails when the new environment is considerably different from the training domain distribution. Because all of the aforementioned works used only hard-contact simulation for training, their performance was highly compromised on soft terrain as the speed of the robot increased. In such conditions, the feet are completely buried into the ground, and rigid-body simulators cannot generate meaningful data for a sim-to-real transfer.

One of the primary reasons for such a trend is a lack of an efficient simulation pipeline for soft and deformable terrain simulation. The leading simulators that these controllers used, such as PyBullet (*21*), MuJoCo (*22*), and RaiSim (*23*), cannot capture the complex dynamics of soft and deformable terrains. They are

Robotics & Artificial Intelligence Lab, KAIST, Daejeon, Korea.
*Corresponding author. Email: jhwangbo@kaist.ac.kr

based on rigid body dynamics or a variation thereof and cannot model the deformation of complex terrains such as a sand beach. Consequently, the necessity of an accurate and fast simulation solution for such terrains arises.

In the granular physics community, there have been multiple efforts to simulate soft substrates. Specifically, granular media (GM)—a collection of discrete solid particles—has been studied to model the naturally occurring soft ground. Unfortunately, a unified solution to simulate intrusions on diverse soft and deformable terrains in 3D has not been found and is still under active research. One of the promising directions for GM simulation is the discrete element method (*24*, *25*), which models individual grains and their interactions. This approach requires a prohibitively long computation time, which makes it inappropriate for RL. Some other branches of terramechanics use reduced-order models, such as the unidirectional spring model (*12*) and viscoplastic model (*11*). They have a lower computation cost, but the results might be substantially different from the reality because they cannot capture the hydrodynamic-like nature of GM.

More promising approaches build on granular resistive force theory (RFT), which is computationally efficient and provides more accurate results in predicting the bulk reaction using empirically derived and experimentally verified models (*26*–*28*). Among them, (*26*, *28*) proposed models describing the force exerted on the surface element and predicted the reaction by integrating force elements over the colliding surface. Although numerical integration can provide an accurate estimate of the net force and torque, summing up all reaction force elements over the submerged area involves a large amount of computation, which may obstruct the application of the data-driven method.

In contrast, the granular cone model proposed by Aguilar and Goldman (*27*) computes a bulk reaction depending on the intruder's kinematics while accounting for transient granular dynamics, which can be salient in the rapid intrusion. The model eliminates the need for integration over surface, offering a computationally efficient way to approximate the granular intrusion as a point interaction. Although the model cannot explain the rate dependency of granular intrusion captured in (*28*), its effect is relatively small in the typical operating range of legged machines.

Despite the computational merit, the granular cone model is deficient in simulating general locomotion because it only addresses the vertical reaction for the intruder of a specific shape. To work around this, we further extended and modified the model to efficiently simulate a general multilegged robot under assumptions admissible for shallow penetrations. Specifically, we approximated the interactions between the nonrigid substrates and the intruder as a point contact and modeled the tangential forces as Coulomb friction. We solved this model using the block Gauss-Seidel method to handle multicontact scenarios. The solver used a filter designed to smooth the normal reaction to prevent a positional drift of the intruder. In addition, we introduced the horizontal stroke–resistive (HSR) force model, which describes the impeding reaction exerted by the GM when the intruder travels horizontally. These modifications allowed us to generate sufficient data while maintaining the desired accuracy for a sim-to-real transfer.

**Movie 1. Agile robotic locomotion of Raibo over deformable terrains.**

However, good simulation alone is not sufficient to dynamically control a legged robot because we do not have accurate information about the terrain properties a priori. These properties cannot be estimated reliably with a vision system because they sometimes do not exhibit distinct visual features. To resolve this unperceivable nature of the terrain properties, we used domain randomization of the terrain properties and an adaptive locomotion strategy that uses haptic information from the feet. The strategy is realized with the proposed control architecture, which processes proprioceptive data describing the robot's internal states. It comprises the recurrent encoding module to compress the sensor observation stream and the explicit estimation module for veiled state variables. The controller encodes the history of the observation in both explicit and implicit ways simultaneously to identify the terrain properties and to adapt the control strategy accordingly. Large-scale randomization on simulated terrain characteristics fosters the discerning ability for surroundings and produces a terrain-agnostic controller adaptable to a broad spectrum of environments.

We followed the presented approach to successfully train a controller for a quadruped robot, Raibo, a dynamic and versatile quadrupedal robot. The robot can run over diverse deformable terrains, including soft beach sand, athletic track, vinyl tile, soft air mattress, and grass field. Compared with the previous controllers, our controller achieves a lower failure rate, higher maximum speed, and lower cost of transport (CoT).

## RESULTS

The presented controller trained on simulated deformable terrains has been deployed on the real robot Raibo (fig. S1) in diverse environments. It controls four legs, each of which contains three joints: hip abduction/adduction, hip flexion/extension (HFE), and knee flexion/extension (KFE). Detailed specifications for our target hardware are provided in section S2. Movie 1 contains the results and a brief explanation of our work, and Fig. 1 shows examples of the test environments.

In the following sections, we used a learned state estimator (*7*) to estimate the state and related properties thereof. The estimation performance can be found in section S4.

### Beach sand

The trained controller demonstrated dynamic locomotion skills over dry and wet beach sand (Movie 1 and Fig. 2). On the dry sand, the feet (6-cm diameter) were sunk entirely into the sand during the stance phase, while the robot was running at high speed. On this terrain, our terrain-agnostic controller was able to achieve forward body speed up to 3.03 m/s and to turn robustly at

**Fig. 1. Quadrupedal robot on various terrain.** The proposed controller was tested on diverse terrains.

a yaw rate of around 0.92 rad/s. The wet sand was generally firmer and required higher stress to yield, showing characteristics close to those of the rigid ground. As shown in movie S1, the controller could continue walking on the terrain transitioning from dry to wet sand, and it achieved a similar maximum speed of around 3.0 m/s after the transition. Traversing along the shore, the robot encountered both inclining and declining slopes up to about 4° and irregular pits and bumps on the surfaces. Although both situations were not modeled in the simulation, the policy could robustly run on those surfaces.

Figure 2 (A and B) illustrates the case when the front-left foot encountered a loosely packed region while the robot was running at a speed of around 2.5 m/s. The robot lost its balance because the sand beneath the front-left foot did not generate a sufficient reaction force. It immediately swung the front-right foot with a high clearance and placed it on the next contact point where the reaction force could be stably generated. This sequence of actions could stabilize the robot in a fraction of a second and enable it to achieve near the commanded locomotion speed.

On the dry beach sand, the proposed method was compared with PL, a state-of-the-art training method for a quadrupedal robot (*5*, *6*, *16*). Using the PL method, we trained two baseline policies as performance references for the presented controller: PL/Env-Rigid and PL/Env-GM. The former was trained in randomized rigid rough terrains, where the hard contact occurs between foot and ground. In contrast, the latter experienced the proposed deformable terrain simulation during training. For training Teacher/Env-GM, the teacher policy of PL/Env-GM, we provided terrain parameters of the simulation as the privileged information.

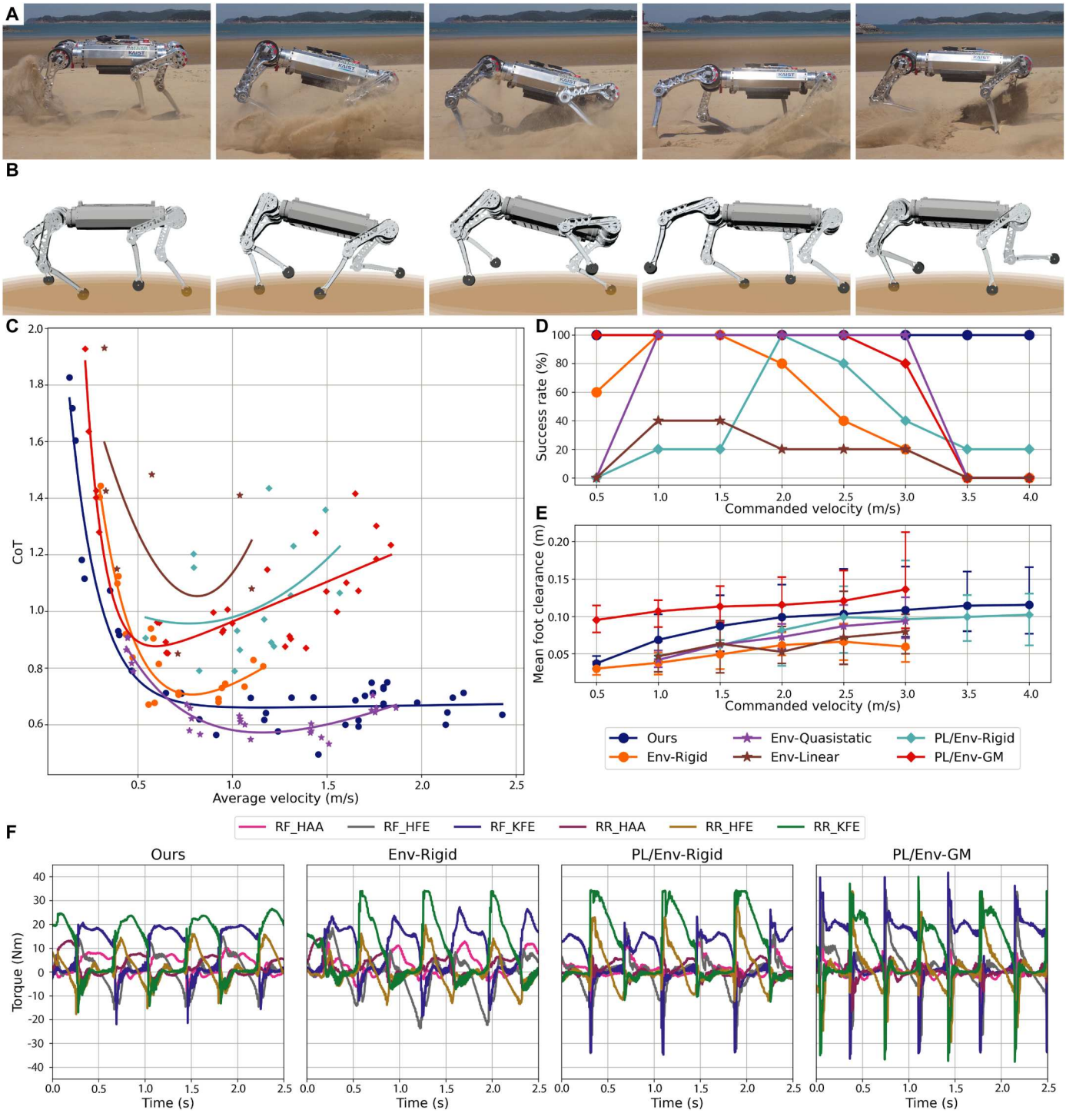


**Fig. 2. Deployment on the soft beach sand.** The presented controller could traverse robustly on dry beach sand despite some unpredictable obstacles. (**A**) A sequence of recovering motions from the loosely packed region suddenly encountered by the front-left foot while running at 2.5 m/s. (**B**) A sequence of the recovery motions reconstructed using state estimation. For (A) and (B), the four right images correspond to 0.27, 0.37, 0.62, and 0.78 s after the time of the leftmost image. (**C**) CoTs for different average velocities and controllers. CoTs were measured, while the robot was walking on a 5-m sand track and presented as dots. Curves result from the nonlinear fitting of $ae^{bx} + cx + b$ to CoTs. (**D**) Success rates for different commanded velocities. Each success rate was evaluated over five trials. (**E**) Foot clearances for different commanded velocities. To measure how high the resultant controllers lifted the feet, we computed the IQRs of the foot heights projected on the sagittal plane. The line graph represents the expectation of IQR over feet and successful trials, and the caps represent the maximum and minimum IQRs among them. (**F**) The measured torques of the two right legs during forward walking (around 1.0 m/s).

To further study the contribution of simulated terrain characteristics, we additionally trained three ablated controllers by replacing the proposed simulation with uneven rigid terrain, the GM simulation without inertial drag forces, and a spring-damper terrain model. We refer to them as Env-Rigid, Env-Quasistatic, and Env-Linear, respectively. For these ablated controllers, training methods other than the simulation were kept the same as the presented, full-featured controller (Ours). In all controller training, the objective defined by the reward function remained unchanged: to track the desired linear velocity and yaw rate of the torso while achieving high robustness and energy efficiency. Detailed descriptions for the baseline and ablated controllers can be found in section S3.

We quantitatively evaluated how learning frameworks and training environments affect the performance of the controllers with respect to the objectives. The policies deployed on the real robot underwent five repetitions of a 5-m flat sand track for each commanded forward velocity ranging from 0.5 to 4.0 m/s. The corresponding results on energy efficiency, success rate, foot clearance, torque usage, and speed can be found in Fig. 2 (C to F) and Table 1. In each experiment, the robot was placed 1 m before the start line. Any body-ground contact or joint position limit violation was considered a failure for safety. Throughout the controller evaluations, we used the safety cable to prevent the hardware from being damaged but took care to keep the measurement unaffected. The foot clearances presented in Fig. 2E were calculated as interquartile ranges (IQRs) of foot heights in the sagittal plane, reconstructed from the torso′s orientation and joint positions.

Figure 2D shows the success rate of the task for each command velocity and controller. On the experimented sand terrain, the presented controller exhibited stable locomotion performance over all command velocities without a single failure. On the contrary, PL/Env-Rigid achieved an overall success rate of only 37.5%. Unexpectedly, the success rate of this controller versus the commanded speed appears as a bell shape in the plot. The policy achieved high success rates for command velocities between 2.0 and 2.5 m/s but low success rates for higher and lower command velocities. The main reason for terminations in the low command range was insufficient foot lifting, which created a stiff sand pile in front of the feet. In the high command velocity range, on the other hand, severe sinking of the rear stance feet often caused the torso to heavily deviate from its standard orientation and eventually led the robot to failure.

PL/Env-GM performed better, achieving an increased success rate of 72.5% and a zero failure rate up to 2.5 m/s command. As shown in Fig. 2E, the policy resulted in much higher foot clearance than the other controllers and safely traversed the sand terrain at low velocities. However, at high command velocities, such motions made the policy susceptible to unmodeled effects. As shown in fig. S5, because the policy has used joint angles close to the limit while producing high clearances, the joint limits were often violated.

Next, we compared the forward locomotion speed achieved by each controller on the sand. The average forward velocity over the 5-m sand track and the top speed after the acceleration phase were measured using a stopwatch and recorded videos. For the latter measurement, the speed was averaged over the last gait cycle before it reached the goal line. Table 1 reports the maximum speed achieved by different controllers. The presented controller outperformed the others by a large margin. Compared with PL/Env-Rigid and PL/Env-GM, the maximum speed of Ours was 83.3 and 50.7% faster, respectively.

Last, the energy consumption during the locomotion of each controller was analyzed. We computed the CoT for each trial as $P/mgV_x$, where the average power $P$ was calculated as a sum of mechanical power $P_{mech} = \sum_t(\boldsymbol{\tau} \cdot \dot{q})\Delta t/T$ and the corresponding rate of heat dissipation $P_{\text{heat}} = \sum_t(\boldsymbol{\tau} \cdot \boldsymbol{\tau}/k_{\text{m}}^2)\Delta t/T$. $\boldsymbol{\tau}$ is the joint torques, $\dot{q}$ is the joint velocities, $mg$ denotes the weight of the robot, $T$ is the time duration from start to the goal line, and $V_x$ is the corresponding average forward velocity. We measured $\boldsymbol{\tau}$ and $\dot{q}$ for time $t$ from motor drives with a period of $\Delta t$. The motor constant $k_{\text{m}}$ was provided by the manufacturer. Figure 2C presents the CoTs versus average velocities for different controllers. The presented controller recorded considerably lower CoTs than the controllers trained with PL. The measured torques from each joint, while the robot was running at the average speed nearest to 1 m/s, are shown in Fig. 2F. For the flexion/extension joints, i.e., KFE and HFE joints, the positive direction is for the extension, and the negative direction corresponds to the flexion. Ours used lower peak torques than the other controllers, which contributes to its low CoTs.

The ablated controllers were evaluated under the same objectives, and the corresponding results are shown in Fig. 2 (C to F) and Table 1. Env-Rigid recorded a 50% overall success rate. Failures occurred at high-speed commands because of similar reasons to PL/Env-Rigid. Meanwhile, although the low foot clearances had caused the foot to move inside the sand, the penetration was not severe, and

**Table 1. Maximum average speed while traversing 5 m and maximum speed after acceleration on beach sand.**

| Controller | | Maximum average speed over 5 m (m/s) | Maximum speed after acceleration (m/s) |
|---|---|---|---|
| **Training framework** | **Trained environment** | | |
| Ours | GM (Ours) | 2.427 | 3.028 |
| | Rigid | 1.163 | 1.234 |
| | Quasistatic | 1.866 | 1.959 |
| | Linear | 1.101 | 1.658 |
| PL | GM (Ours) | 1.838 | 2.009 |
| | Rigid | 1.567 | 1.652 |

the policy could succeed in the task at low speeds. To measure the penetration, we alternatively calculated the average rate of mechanical work done by leg extensions, $P^E_{\text{mech}} = \sum_t([\tau_{\text{FE}}]^+ \cdot [\dot{q}_{\text{FE}}]^+)\Delta t/T$, because most of the energy was consumed by the sand during penetration. $[\mathbf{x}]^+$ denotes an element-wise operation, $\max(0, \mathbf{x})$, and $\tau_{\text{FE}}$ and $\dot{q}_{\text{FE}}$ are joint torques and velocities for flexion/extension joints, respectively. $P^E_{\text{mech}}$ was 52.5% larger for the PL/Env-Rigid than the Env-Rigid when the average speed was around 1 m/s (Fig. 2F), which suggested a shallower penetration depth of the latter for low-speed locomotion. Env-Quasistatic could complete the task with commands from 1.0 to 3.0 m/s. However, it failed to control the robot safely at velocity commands higher than this range. Although the safety was compromised, the policy achieved a slightly higher energy efficiency than the full-featured controller there. Third, substituting the proposed simulation with the randomized spring-damper terrain made the controller unstable on the yielding terrain. In most cases, Env-Linear did not generate sufficient torques to walk and let the torso fall on the ground. The resulting success rate of 17.5% was the lowest among all alternatives that we tested.

So far, we have quantitatively analyzed the performance gaps among various controllers on the forward locomotion. However, this does not fully reflect the actual performance gaps that we have encountered during experiments. The controllers not trained in the deformable terrain simulation were very unstable, and terminations often occurred immediately after the initialization or when the robot was walking back to the starting position for a reset. All such scenarios were not recorded as a failure. Furthermore, failures of those trained on the rigid ground or spring-damper system often involved fiercely oscillating motions, which could damage the robot. The presented controller was the only one that could perform all tasks successively without a system reinitialization. We did not observe any unstable oscillations throughout the experiments.

### Various terrains

We further investigated the performance of the presented controller on vinyl tile, athletic track, grass, and an air mattress, as shown in Fig. 3 (A to C and H), to test the controller′s adaptability to different ground stiffness. The vinyl tile–covered cement floor was the closest to the ideal rigid ground, having negligible compliance. The porous athletic track was made of polyurethane-bound rubber granules and had lower surface stiffness than the vinyl tiles. The grassy terrain was even softer than the athletic track, and its intrusion dynamics was noticeably different from that of GM. The air mattress allowed much deeper penetrations but differed from our GM simulation because the reaction forces are exerted during lifting as well. In addition, the reaction forces on each foot are highly coupled through the air pressure, and the control problem becomes more challenging.

Although none of the tested terrains was simulated during training, the presented controller could generalize to the tested terrains: The robot could achieve forward locomotion speed up to 3.13, 3.62, and 3.76 m/s on grass (movie S2), athletic track (movie S3), and vinyl tile (movie S4), respectively. In addition, in all terrains, the robot stably maintained the turning speed of around 0.94 rad/s. Because of the size limitation, we tested only yaw turning on the air mattress, as shown in movie S5. The presented controller could maintain balance and turn at 0.71 rad/s under the command of 1.0 rad/s and at 1.54 rad/s under the command of 2.0 rad/s on the air mattress.

On the vinyl tile, running track, and grass field, we quantitatively compared the presented controller with Env-Rigid. To this end, we set up a 5-m straight running track on each terrain and repeated the same running test done on the sand. Measurements were obtained the same way, except that we initialized each controller at least 5 m before the start line to evaluate them after a sufficient acceleration period. The corresponding results are shown in Fig. 3 (D and E).

On all three terrains, both controllers showed comparable locomotion speeds up to the commanded speed of 3.0 m/s, as shown in Fig. 3E. At higher command velocities, however, Env-Rigid experienced substantial degradation in performance on the athletic track and grass. It showed 6.1 and 10.1% lower maximum speed than our controller on the running track and grass, respectively. In particular, Env-Rigid recorded an apparent decline in speed when the command was increased to 3.5 m/s on the grass. In this case, we found that the policy produced an abnormally large foot clearance (fig. S6), and, consequently, the running speed was lowered. The same problem drove the robot to fail at the command at 4.0 m/s. On the vinyl tile, we could not observe a meaningful difference between the policies.

Next, we compared the CoTs of the controllers on multiple terrains as illustrated in Fig. 3D. On all terrains, including the beach sand (Fig. 2C), the presented controller exhibited comparable or lower CoT than Env-Rigid. However, the CoT increased steeply at high speeds on the running track and grass. We hypothesize that the inaccuracies of the proposed model in simulating the nongranular terrain caused the policy to act more conservatively with higher foot clearances (fig. S6). More detailed descriptions of this phenomenon are provided in section S5. On all terrains, the CoTs of the presented controller were less sensitive to the ground softness than the CoTs of Env-Rigid.

To visualize how the presented controller encodes the differences of the terrains in the latent space, we conducted $t$-distributed stochastic neighbor embedding ($t$-SNE) on the hidden states $\mathbf{h}_t$ condensed by the recurrent encoder. Figure 3F shows the result of $t$-SNE for the commanded controller (1 m/s) deployed on the above three terrains and the dry sand. We found that embedded hidden states distributed distinctly for different terrains and thus carried sufficient information about the terrain type. The band shape of the distribution for each terrain represents the cyclic nature of the gait, as shown in fig. S7.

On the air mattress, the yaw rate of 1 rad/s was commanded after the robot was initialized on the mattress, as shown in Fig. 3G. We considered more than a half turn a success, and five clockwise and five counterclockwise trials were conducted. To compare the energy efficiency of different controllers, we defined the cost of turning (CoTu) as $P/mgl_c\omega_z$, which is analogous to the CoT for turning motions. $\omega_z$ is a yaw rate, and $l_c$ represents the characteristic length, which was set to the default distance between two diagonal feet. Four controllers were deployed in this setting: Ours, Env-Rigid, PL/Env-GM, and PL/Env-Rigid. Results are shown in Fig. 3H.

The presented controller was the only one that could perform five turns in a row in each direction. The other controllers exhibited very unstable motions on this terrain. In particular, PL/Env-Rigid could not make even a half turn. Env-Rigid and PL/Env-GM could

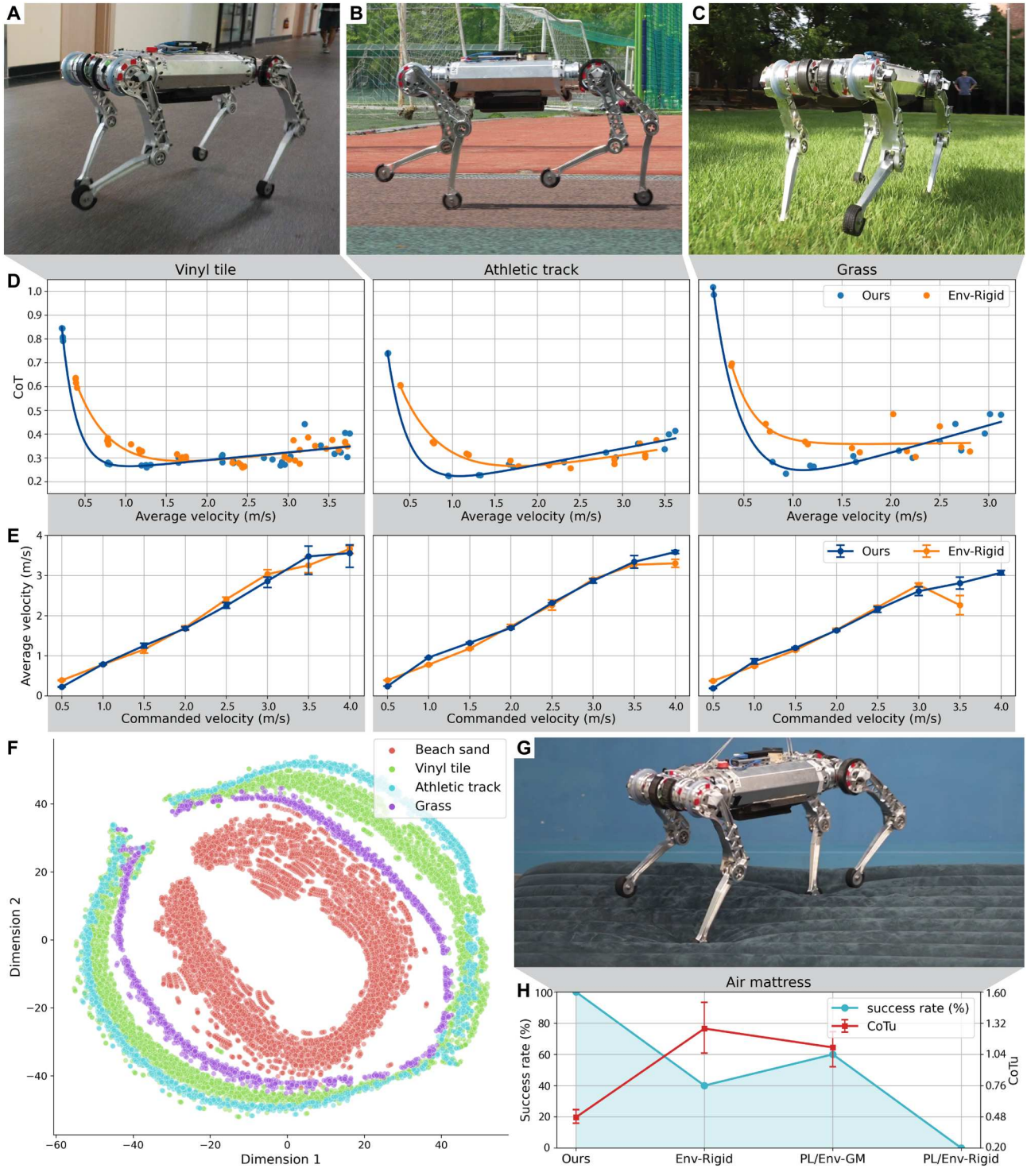


**Fig. 3. Evaluation on various terrain.** The presented controller was evaluated on vinyl tile (**A**), athletic track (**B**), grassy terrain (**C**), and an air mattress (**G**). For the former three terrains, the presented controller was compared with the one trained with the rigid contact. (**D**) CoTs for different average velocities on each terrain. Dots represent the measured CoTs, and curves result from the same fitting with Fig. 2D. (**E**) Measured average velocities against the commanded velocities on each terrain. Caps represent the minimum and maximum speeds given the velocity command. (**F**) The *t*-SNE visualization for the encoded latent vectors that were collected on the different terrains using the presented controller. (**H**) Success rates and CoTus for yaw turning on the air mattress. The success rate was computed from five positive and five negative turns, and the caps represent the maximum and minimum CoTus over successful trials.

perform more than half a turn but at a much lower energy efficiency than Ours. This test and the results are illustrated in movie S5.

We further evaluated the presented controller's ability to adapt to different terrain characteristics during locomotion. In this experiment, the controller encountered a sudden terrain transition from a rigid brick road to a soft memory foam mattress and could stably traverse over it with a command at 0.5 m/s, as shown in movie S6 and fig. S8. In-depth analyses of this experiment are available in section S6.

### Simulated deformable terrains

We deployed the controllers on the proposed GM simulation to meticulously investigate the behavioral adaptation under the terrain variation. Test environments were constructed from an equally spaced grid of two stiffness parameters, flat-surface resistive stress ($\sigma_{flat}$), and conical-surface resistive stress ($\sigma_{cone}$), where the ranges were 1.0 to 10.0 MN/m $^3$ and 0.15 to 0.6 MN/m $^3$, respectively. The other parameters were kept constant at default values presented in table S3. For each environment, controllers were tested 360 times with a randomly sampled velocity command for 4 s.

In these simulated environments, we evaluated the policies trained with the proposed simulation or hard contact with reward function and foot clearance to see how locomotion performance and walking behavior change under different terrain characteristics. The clearance was calculated as an average foot height from the mean penetration depth of stance feet. Figure 4 (A to D) and table S1 show the results. Env-Rigid and PL/Env-Rigid showed miserable reward degradation on compliant grounds and did not display a noticeable adaptive behavior in clearance. The foot clearances of Ours and PL/Env-GM adapted flexibly to terrain characteristics (Fig. 4D). Meanwhile, Ours outperformed PL/Env-GM in terms of reward for all conditions tested and used lower foot clearance, which indicates efficient motion. Section S7 provides detailed measurement methods and comparisons.

We explored an alternative architecture for the encoder from the same simulation settings. We replaced the recurrent encoding module with a time-convolutional network and examined the resulting policy. The recurrent module exhibited a higher overall reward, and the policies with a time-convolutional network did not display a noticeable adaptation to each test terrain. Figure S10 shows the results, and corresponding descriptions are available in section S8.

We next studied how much each input affects the output in the multilayer perceptron (MLP) actor module included in a trained controller. In the presented controller, the actor determines the subsequent motion while taking the desired velocity command cmd, the encoded observation history $\mathbf{h}_t$, and the estimated states $\mathbf{e}_t$. Instead of directly examining it, we trained oracle policies, which are policies trained with additional inputs unobservable in the actual application, for further analyses. Oracle(H) policy has the actor that takes parameters for the deformable terrain simulation $\mathbf{\Theta}_T$ in addition to our proposed scheme. Oracle(O)'s actor module similarly takes $\mathbf{\Theta}_T$, yet $\mathbf{h}_t$ is replaced by the current observation $\mathbf{o}_t$ of current sensor data. Detailed descriptions are provided in section S3. We evaluated the above oracles and Teacher/Env-GM, which also has direct access to $\mathbf{\Theta}_T$, with the same metrics as the other policies. In addition, to understand the input-output relation of the actor modules, we conducted attribution analyses on Oracle(H) and Oracle(O). We simulated each policy for 4 s in 10,800 randomized environments as we did for the training (sections S15 and S16) and calculated integrated gradients (*29*) and group feature ablations on the actor modules. An averaged input was used for the reference for calculating them, and we took only the magnitudes of the attributions to represent the sensitivity of the output to the input. The results are presented in Fig. 4 (E to G).

The attribution results implied that a module performing explicit state estimations was the key feature that contributes substantially to the final performance. As shown in Fig. 4E, for both policies, the attribution calculated by group feature ablation was much higher for the estimation group $\mathbf{e}_t$ compared with the terrain parameter group $\mathbf{\Theta}_T$. Such a high dependency on $\mathbf{e}_t$ suggested that the estimated features provided the network with decisive features in determining actions. Despite the privileged access to the ground-truth features, Teacher/Env-GM could not reach the reward of the presented controller and the other two oracle policies (Fig. 4, A and B). These observations demonstrated the efficacy of a concurrently trained state estimation module.

Another feature of the proposed method is the implicitly encoded observation history, transferred to the actor along with the state estimations. Among Oracle(H) and Oracle(O), the one with the encoded history outperformed the other by a meaningful margin. Oracle(H) exhibited the most skilled locomotion among all policies, as presented in Fig. 4 (A and B). Removing the encoded history from the input resulted in degeneration and recorded a similar mean reward with the presented policy (Ours).

Oracle(H) manifests less attribution for the terrain parameters than Oracle(O), as shown in Fig. 4E. In addition, the attribution of terrain parameters was generally more sensitive to some parameters that were used for the simulation when the encoded history was absent, as presented in Fig. 4F and fig. S11. These findings suggested that, when both the encoded history and the true terrain parameters are given to the actor, the actor considerably relied upon the former to get terrain-specific information.

In the last part, we investigated integrated gradients of each terrain parameter from the same simulation for Oracle(O) to study how much each terrain characteristic affects the following action. Figure 4G shows the result. It revealed that the depth-dependent parts of the ground reaction force corresponding to parameters $\sigma_{flat}$ and $\sigma_{cone}$ were more substantial than the other parts, such as the inertial force of the ground. Section S9 provides more analyses.

## DISCUSSION

The presented controller demonstrated a high level of agility and robustness on various deformable terrains. Our controller stably drove the quadrupedal robot over dry beach sand even when encountering unexpected obstacles such as slopes and loosely packed regions during high-speed locomotion. It substantially outperformed the baseline and ablated controllers regarding success rates, maximum speed, and energy efficiency at all velocity commands. The learned locomotion skill could also adapt to artificial environments or nongranular substrates, although the training environment was confined to a granular model. Its performance was relatively insensitive to wide terrain variations and could generalize to an air mattress that requires fastidious control.

The ability to acclimate to different ground characteristics primarily emanates from the appropriate terrain distribution for training. Training policies without ground deformation resulted in

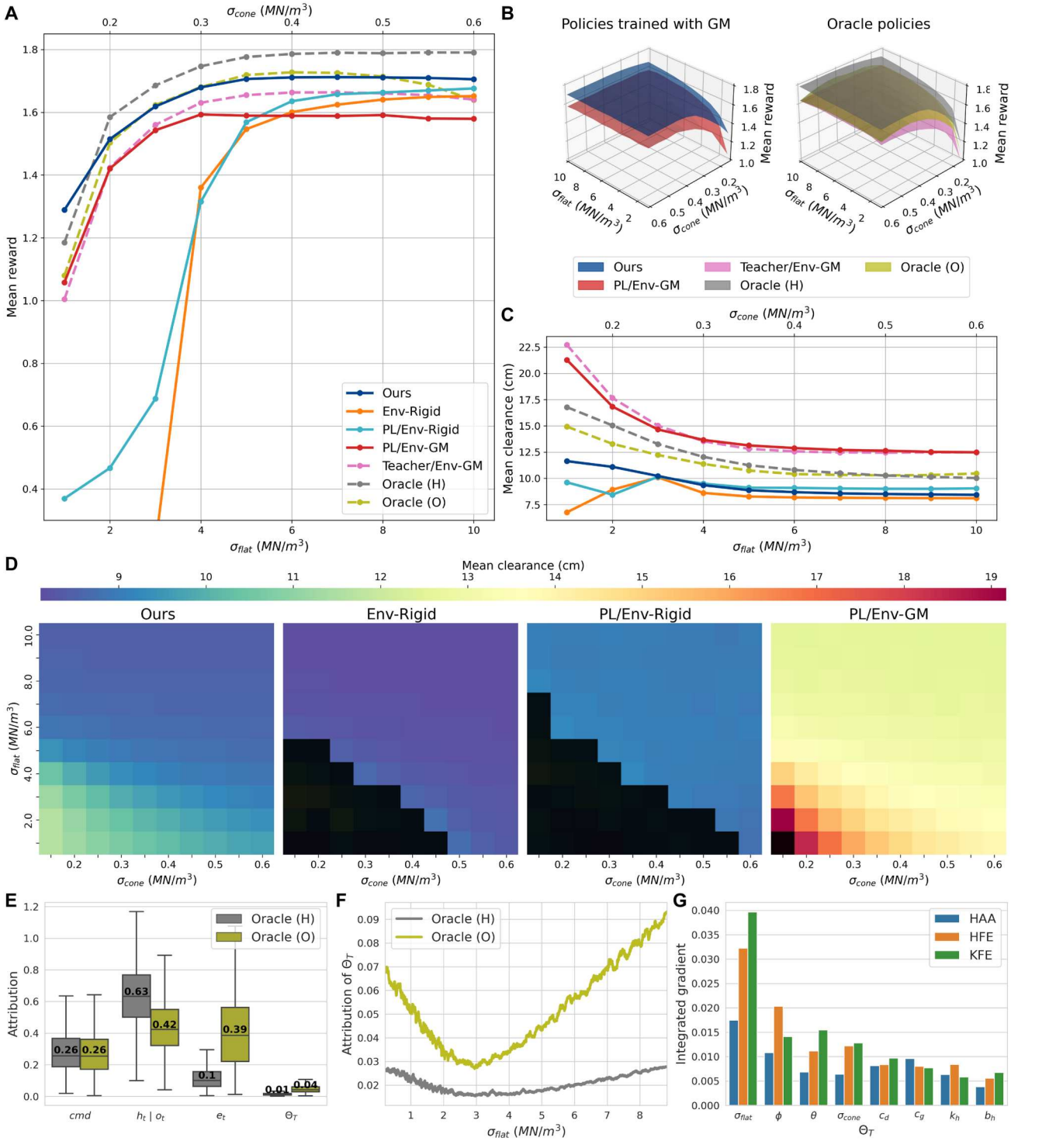


**Fig. 4. Evaluation in simulation.** The controllers were further quantitatively evaluated in the proposed deformable terrain simulation. (**A** and **B**) Mean rewards computed with eq. S17 for different ground stiffnesses. (**C** and **D**) Mean foot clearances for different ground stiffnesses. They are computed as a time average of the foot height with respect to the average maximum penetration of stance feet $z_g$. The black area represents parameter sets that could not achieve a higher success rate than the criteria, 10%. (**E**) Group feature ablation results for each group provided to the two oracle policies. (**F**) Group feature ablation of terrain parameter $\Theta_T$ for different stiffness parameter $\sigma_{flat}$ used for the simulation. (E) and (F) only present the magnitude of each attribution to represent the output's sensitivity to the input. (G) Averaged magnitude of integrated gradient attributions on each terrain parameter for three different joints of the legs.

apparent degradation in performance as the test terrain softness increased. Notably, such degeneration occurred even on the athletic track, where the deformation appears to be negligible. Even when the ground compliance was tackled, neither relying on a simple spring-damper model nor taking only the quasistatic forces could enhance the controller much on the deformable terrain. Our method, established on the appropriate terrain interaction physics, could well approximate the reality, resulting in a performant controller.

Before this work, simulation techniques for compliant and yielding terrains were not fast enough for RL because they primarily focused on accuracy. Because the computation time made them impractical options, earlier research works had to use rigid ground simulators (*5*). The proposed methodology enables simulating diverse compliant contact efficiently with admissible accuracy and permits the agent to experience relevant contact scenarios. Consequently, our model tightens the gap between the feasible simulation for training and the physical world so that the RL agent can cultivate adaptability to various nonrigid terrains.

In addition, this work found that a complementary relationship between the contact model and the training framework could be established through domain randomization. The randomization of terrain parameters compensated for the imperfections in the contact model. Diverse contact scenarios roughly close to the actual reaction of the deformable terrain benefited a resulting controller to generalize well across diverse nonrigid terrains without knowing their properties a priori. We see that the approach is not limited to our application but can be used in other practices where the accurate prediction is too costly.

The suitable policy architecture for traversing deformable terrain was also essential. Simulation results and the attribution study revealed the efficacies of having distinct input groups of our controller: the explicit state estimations and the implicit history encoding. In the case of latter, we found that a trained policy considerably depended on the encoded history even when the terrain parameters were explicitly given. This means that our recurrent encoding structure effectively condensed the temporal data such that the subsequent actor module could use them easily.

Our recurrent control architecture outperformed the common PL by a wide margin. The performance of the student PL policy was upper-bounded by the teacher policy in an imitation learning setting. On the other hand, our training pipeline is end-to-end and does not suffer from such limitations. Furthermore, our implicit encoding of history conveyed well the terrain-related information (Fig. 3F), and the resulting controller performed better than the teacher policy Teacher/Env-GM.

There are limitations left to be addressed for future work. We assumed that the contact occurs at a point for fast calculation, and the contact model could admirably approximate the reality in the regime where the intruder is a small convex body and the penetration is shallow. However, if the colliding area between the intruder and the ground increases, the violation of the assumption would severely lower the model accuracy. For applications beyond the current regime, such as flat-plate intruders on the sandy terrain, the contact model should address multiple points or surfaces to maintain accuracy.

Another direction is to widen the model's capability to simulate various terrains. Although the proposed simulation provided fruitful experiences for the policy, we found some type of terrains where the proposed method yielded slightly higher CoT (Fig. 3D), as discussed in section S5. Enlarging the model capability to other environments, such as grass that has firm ground below, will remedy this problem. In addition, the proposed model is currently insufficient to simulate sloped ground, and the consequences were often observed in the outdoor sandy terrain. To this end, we plan to build a model that predicts granular flow and computes the dynamic terrain normal vector accordingly. Last, our model has to be extended to include propulsive and impeding forces while the feet move upward, which can be observed in many types of nongranular terrains such as mudflats and swamps.

The presented controller is still blind—it is not using exteroceptive sensors. Although visual features do not convey sufficient information in most cases, they can be combined with the proposed work to further improve the locomotion performance. The fusion of haptic and exteroceptive information for terrain classification can be very challenging because the spatial correlation of the terrain properties has to be taken into account. However, we believe that such a fusion can lead to animal-like locomotion skills in broader wild environments.

## MATERIALS AND METHODS

### Simulation

Our work focuses on modifying the contact solver and the underlying contact model to simulate various types of deformable ground. The proposed model and solver find contact forces for each colliding point of the system. Our model has its basis in the model proposed by Aguilar and Goldman (*27*), but we introduce several simplifications to make the simulation fast and stable. The resulting contact solver implemented on RaiSim (*23*) enables an RL agent to experience a wide range of terrain characteristics, whereby a robust locomotion policy can be trained. We provide a generic method for robot simulation in section S10, and the following sections describe contact with deformable terrain.

### Terrain model

The model introduced by Aguilar and Goldman (*27*) focuses on the cone-shape jammed grains beneath the intruder to account for the quasistatic and inertial drag force, where the latter comprises a term proportional to squared velocity and an added-mass effect of GM (*30*, *31*). It produces a reaction force depending on the accumulation dynamics of the growing cone of densely packed grains, as illustrated in Fig. 5A. The model explains the depth dependency of the granular stiffness by splitting the stress on the conical and flat areas. Section S11 includes explanations and implementation details for the terrain model, including the following modifications.

Because the original model only explains the interactions of cylindrically shaped objects with GM, it cannot explain the reaction forces on arbitrarily shaped robot feet. Therefore, we assume that intrusions of an arbitrary-shaped intruder would form a similar jamming cone within the predictable range, as discussed by Aguilar and Goldman (*27*). We substitute the top radius of the developing cone with the hydraulic radius of the cross-section $A_{\mathrm{sub}}$ of the intruder, i.e., the intersecting area at $z = 0$. Although this might lead to a compromise in simulation accuracy, our goal is not to model the terrain exactly; it is to match the distribution of the models with the reality such that our randomized terrains can produce a realistic data distribution for RL. Therefore, we

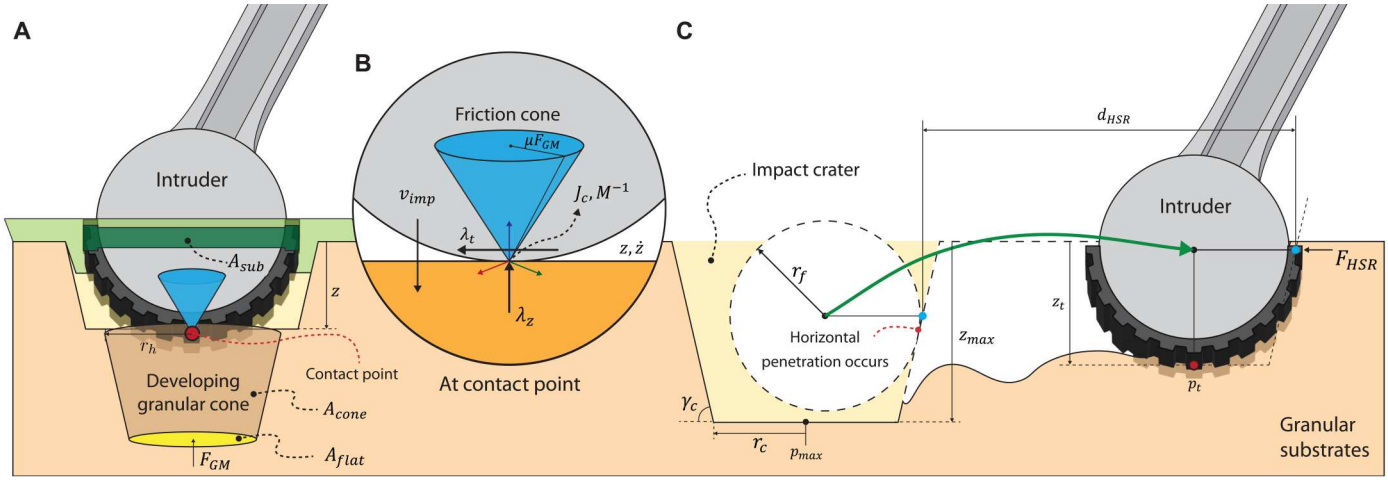


**Fig. 5. Contact model definition.** (**A**) The terrain model predicts a vertical component of a ground reaction force on the basis of the intruder's penetration depth and velocity. The calculation involves the developing granular cone beneath the intruder, as Aguilar and Goldman (*27*) proposed. (**B**) The surface contact between the intruder and the adjacent substrates is approximated as a point contact at the deepest point. The bulk tangential force from the substrates is assumed to be Coulomb friction. (**C**) The HSR force model is introduced to simulate the reaction from the substrates when the intruder moves horizontally in the substrates. The force is computed on the basis of the travel distance $d_{HSR}$ and the current penetration depth $z_t$.

randomize the model parameters such that the ranges of sinkages and slippages in simulation match those in reality.

To further mimic the deformation characteristics often found in natural soft substrates, we assume that a penetration leaves the ground deformed and that the ground cannot exert force while the intruder moves up. Our model simulates these by conditioning the force to be exerted only on the downward locomotor and absent during upward movement.

## Contact model

To rapidly simulate the interactions with soft substrates, we approximate the surface contact between a foot and the deformable terrain as a point contact. In other words, we define a contact model to make the resultant force of the terrain model apply to a single contact point. The point is assumed to be the deepest spot of the plunged intruder. Because the terrain model calculates the normal reaction depending on the kinematics, not the shape of the colliding surface, it is suitable for point contact approximation. On the other hand, this conversion yields several problems. First, there is no existing model describing the tangential force for a point contact. The granular cone model describes only the force in the normal direction, and other models based on surface contact (*26*, *28*) require integration to get the sum of horizontal forces exerted on the surface elements.

We assume the Coulomb friction at the contact point between the ground and the intruder to model the tangential force (Fig. 5B). We pose that single Coulomb friction calculated from the resultant lift force can produce enough approximate experiences for training. This assumption reduces the computational complexity, which is beneficial for our data-driven method. We exclude the inertial force terms when computing the tangential forces to further reduce the errors due to the point contact assumption. Agarwal *et al.* (*28*) proposed a surface contact model where the inertial force proportional to the squared velocity is only exerted in the surface-normal direction. Because the inertial force is typically the biggest just near the ground surface, including it to produce the friction force will yield an inordinate tangential drag there and result in a considerable dissimilarity with the surface contact model. Rather than taking this difference, we randomized the friction coefficient so that the trained policy can learn robust behaviors. Our model also does not consider static friction because the substance easily yields and flows.

In addition, to capture the resistive force when the buried foot is dragged horizontally, we introduce the HSR force model (Fig. 5C). It assumes the impact crater as a truncated cone centered at the point where the maximum penetration occurs and the intruder shape as a sphere of a radius $r_f$. Having the height of $z_{\max}$, the cone geometry is parameterized with the radius of the bottom circle $r_c$ and the incline angle $\gamma_c$.

The HSR force model defines the planar reaction force exerted on the intruder when it moves horizontally beyond the crater cone boundary. It uses a simple spring-damper–like model to represent HSR force $\boldsymbol{F}_{\mathbf{HSR}}$ based on the horizontal travel distance $d_{\mathrm{HSR}}$ as

$$d_{\mathrm{HSR}} = \| \mathbf{p}_t - \mathbf{p}_{\max} \| + \frac{r_f}{\sin \gamma_c} - r_c - \frac{z_{\max} - z_t + r_f}{\tan} \quad (1)$$

$$\boldsymbol{F}_{\mathbf{HSR}} = k_h(d_{\mathrm{HSR}} + \beta_d z_t) + b_h \dot{p}_t \quad (2)$$

where $k_h$ is HSR stiffness, $\mathbf{p}_t$ is the current planar position of the foot, $b_h$ is HSR damping factor, and $\beta_d$ is the depth dependency scaling factor. $\mathbf{p}_{\max}$ is the position of the crater center at the start of the penetration. The reaction force $\boldsymbol{F}_{\mathbf{HSR}}$ increases as the foot moves deeper into the ground and further away from $\mathbf{p}_{\max}$. The force starts acting on the lateral side of the foot against the intruding direction as it penetrates the ground. Considering the plastic deformation on the surface where the horizontal penetration already took place, the force is exerted on the intruder only when $d_{\mathrm{HSR},\ t} > \max d_{\mathrm{HSR},\ t_0 : t-1}$, where $t_0$ represents the time when the horizontal penetration starts. The point of application is located in the intruder's moving direction. We also randomize the stiffness and the

damping factor so that the resulting terrains have a realistic distribution.

The last issue we address in our model is a drifting phenomenon, which makes the intruder slide on the horizontal plane. Because it originates from the integration error, we use an exponential moving average (EMA) filter for the normal force resulting from the terrain model. Details on this phenomenon and the filter design are available in section S12.

## Contact solver

We used the projected Gauss-Seidel solver to calculate impulses on each contact instance of the system. Algorithm S1 describes the solver processing $N$ contact instances on the robot. The solver searches for impulses from multiple contacts by updating each contact in turn. The tangential impulse is initialized with a zero vector, and the normal impulse is fixed to the output of the EMA filter. For each update of impulses, the relative contact velocity of the next time step is predicted and used to correct the tangential impulse. Before accepting the new value, the solver checks the friction cone constraint and projects the tangential impulse to the cone's surface if the constraint is violated. The solver iterates updating the impulses until the total amount of an update converges under the predefined threshold $1 \times 10^{-5}$ N·s. Further details are available in section S10.

A good granular model for RL should be fast and able to simulate diverse contact scenarios at the same time. In this light, we compared the proposed method with dynamic RFT (DRFT) (*28*) by dropping cylindrical objects on each terrain model with randomized parameters. We describe in detail the experiment setup and results in section S13. Our simulation method was about 20 times faster than the DRFT method and made the final positions of the cylinders more widely distributed, as shown in fig. S12. We could not train a controller with the DRFT model because of its prohibitively expensive computation cost. However, we argue that the controller trained with our model will generalize to a wider range of terrains because our model can generate more diverse experiences.

## Training

We used the proposed contact model and solver to build various granular terrain environments with randomized parameters, ranging from ceramic-level stiffness to beach-sand softness. The locomotion problem is formulated as a Markov decision process, as described in section S14. In what follows, we describe details of the training method.

## Controller structure

One of the biggest challenges in such environments is state estimation. Because most state estimators for legged robots rely on a binary contact state, they are compromised in deformable terrains. To this end, we used the learned state estimator presented by Ji *et al.* (*7*), which is trained concurrently with the actor. In this way, the estimator network can predict the terrain deformation and adjust its state estimates accordingly.

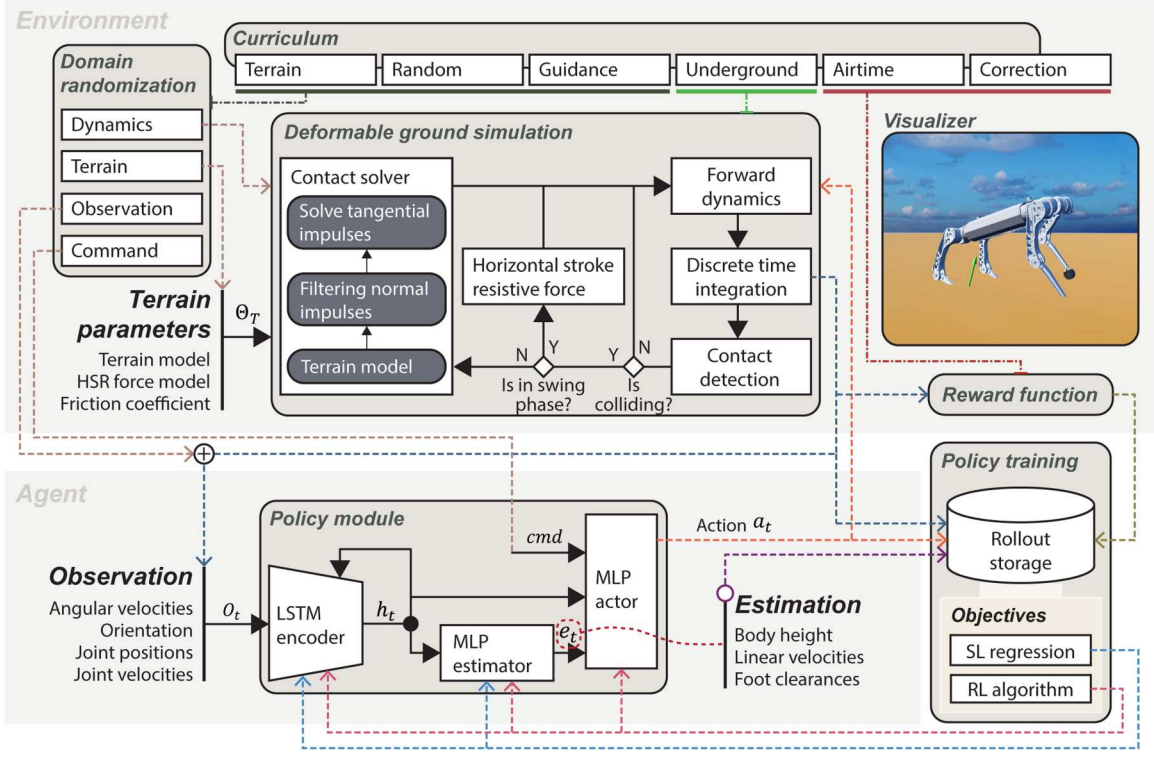


**Fig. 6. Overview of the simulation, training, and control method.** The training uses the proposed deformable terrain simulation. The policy module maps the current observation, and the desired velocity command to the PD targets for the low-level controller. Given the policy output, the simulation calculates the next state of the robot. During the interactions between the environment and the agent, rewards and ground truths for estimations are stored. An RL algorithm and a supervised learning (SL) objective are used to update the policy module. We apply the domain randomization and curriculum learning to facilitate the training.

Our overall control architecture is shown in Fig. 6. The controller comprises a long short-term memory (*32*) encoder, MLP estimator, and MLP actor. The encoder observes the current sensor data $o_t = (\boldsymbol{\omega}_t, \psi_t, \mathbf{q}_t, \dot{q}_t)$, where $\boldsymbol{\omega}_t$ is the angular velocity, $\boldsymbol{\psi}_t$ is the base orientation, and $\mathbf{q}_t$ and $\dot{q}_t$ are the joint position and velocity. A recurrent structure helps form a compact representation for the history data, which is essential for estimating the terrain properties. The output encoded vector $\mathbf{h}_t$ is fed to the MLP estimator, which predicts the body height $z_{\text{body}}$, the foot heights $\mathbf{z}_f$, and the linear velocity of the robot torso with respect to the body frame $\mathbf{V}$. Given the predicted state and the encoded history, the actor network takes a concatenated vector $\mathbf{i}_t = (\mathbf{cmd}, \mathbf{h}_t, \mathbf{e}_t)$ and produces an action $\mathbf{a}_t$ that specifies the joint position targets at 100 Hz. **cmd** represents a 3D vector of the desired velocity, having two elements for the desired translational velocity and one for the desired yaw rate, and $\mathbf{e}_t$ is an estimation vector $\mathbf{e}_t = (z_{\text{body}}, \mathbf{V}, \mathbf{z}_f)$. Last, the low-level positional-derivative (PD) controller tracks $\mathbf{a}_t$ at 4 kHz with fixed gains and zero joint velocity target. Implementation details are provided in section S3.

We set several important kinematic states to be estimated, but the parameters for the simulation were excluded. It is challenging to infer the terrain parameters from the proprioceptive observation alone because their contributions to the granular reaction force overlap and act synthetically. Instead, we provide an encoded vector to the actor network, enabling it to take account of terrain characteristics implicitly. This approach can be considered a mixture of providing an actor with a latent vector that contains the history information (*5*, *6*, *16*) and using the explicit state estimation module (*7*).

### Training in simulation

During training, the algorithm gathered the 4 s of transition data from 360 parallel simulation environments to train the encoder, estimator, and actor networks. We used proximal policy optimization (*33*) to train the networks, and the estimator and encoder were further subjected to supervised learning. Supervised learning uses the mean squared error with the ground truth from the simulation as a loss function. The net training time for the presented policy was about 50 hours.

To facilitate the training, we manipulate the initial state distribution to resemble practical scenarios. We make the environment reset in two ways: hard and soft reset. The former simulates the initialization of the robot in a new environment, and we harshly disturb the robot state for this case. On the other hand, the latter represents the case of terrain transitions during control. With this condition, the robot inherits the previous states. Further details for the environment resets are provided in section S15.

We extensively use domain randomization techniques for the two goals: a seamless sim-to-real transfer and applications in diverse environments (*19*, *34*). We broadly randomized terrain parameters so that each simulated agent underwent different intrusion scenarios, and the abundant terrain-foot interactions could be collected. The randomization ranges are presented in table S3. We also noisify the observation to address the sensor noises and uncertainties in the dynamics parameters, such as kinematic positions of the feet, to make the controller more robust. Section S16 describes how each component is randomized.

Reward functions are formulated such that the agent gets a high reward as it tracks the commands with energy-efficient and robust motions. It does not specify the gait pattern as trot, although some components prompt symmetric locomotion behaviors. We also use curriculum learning to induce desired behaviors relatively quickly. Reward functions and the curriculum are described in detail in sections S17 and S18, respectively.

## Supplementary Materials

**This PDF file includes:**
Sections S1 to S18
Figs. S1 to S12
Tables S1 to S4
Algorithm S1
References (*35*–*39*)

**Other Supplementary Material for this manuscript includes the following:**
Movies S1 to S6

## REFERENCES AND NOTES


1. H. Kolvenbach, P. Arm, E. Hampp, A. Dietsche, V. Bickel, B. Sun, C. Meyer, M. Hutter, Traversing steep and granular martian analog slopes with a dynamic quadrupedal robot. arXiv:2106.01974 (2021).
2. P. Bergmann, K. J. Pettinelli, M. E. Crockett, E. G. Schaper, It's just sand between the toes: How particle size and shape variation affect running performance and kinematics in a generalist lizard. *J. Exp. Biol.* **220**(Pt 20), 3706–3716 (2017).
3. N. Mazouchova, N. Gravish, A. Savu, D. I. Goldman, Utilization of granular solidification during terrestrial locomotion of hatchling sea turtles. *Biol. Lett.* **6**, 398–401 (2010).
4. H. Marvi, C. Gong, N. Gravish, H. Astley, M. Travers, R. L. Hatton, J. R. Mendelson III, H. Choset, D. L. Hu, D. I. Goldman, Sidewinding with minimal slip: Snake and robot ascent of sandy slopes. *Science* **346**, 224–229 (2014).
5. J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, Learning quadrupedal locomotion over challenging terrain. *Sci. Robot.* **5**, eabc5986 (2020).
6. A. Kumar, Z. Fu, D. Pathak, J. Malik, RMA: Rapid motor adaptation for legged robots, in *Proceedings of the Robotics: Science and Systems* (2021).
7. G. Ji, J. Mun, H. Kim, J. Hwangbo, Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion. *IEEE Robot. Autom. Lett.* **7**, 4630–4637 (2022).
8. H. W. Park, P. M. Wensing, S. Kim, High-speed bounding with the MIT Cheetah 2: Control design and experiments. *Int. J. Robot. Res.* **36**, 167–192 (2017).
9. C. Li, P. B. Umbanhowar, H. Komsuoglu, D. E. Koditschek, D. I. Goldman, Sensitive dependence of the motion of a legged robot on granular media. *Proc. Natl. Acad. Sci. U.S.A.* **106**, 3029–3034 (2009).
10. C. M. Hubicki, J. J. Aguilar, D. I. Goldman, A. D. Ames, Tractable terrain-aware motion planning on granular media: An impulsive jumping study, in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2016), pp. 3887–3892.
11. V. Vasilopoulos, I. S. Paraskevas, E. G. Papadopoulos, Compliant terrain legged locomotion using a viscoplastic approach, in *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2014), pp. 4849–4854.
12. D. J. Lynch, K. M. Lynch, P. B. Umbanhowar, The soft-landing problem: Minimizing energy loss by a legged robot impacting yielding terrain. *IEEE Robot. Autom. Lett.* **5**, 3658–3665 (2020).
13. A. H. Chang, C. M. Hubicki, J. J. Aguilar, D. I. Goldman, A. D. Ames, P. A. Vela, Learning terrain dynamics: A gaussian process modeling and optimal control adaptation framework applied to robotic jumping. *IEEE Trans. Control Syst. Technol.* **29**, 1581–1596 (2021).
14. J. Siekmann, K. Green, J. Warila, A. Fern, J. Hurst, Blind bipedal stair traversal via sim-to-real reinforcement learning, in *Proceedings of the Robotics: Science and Systems* (2021).
15. J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, M. Hutter, Learning agile and dynamic motor skills for legged robots. *Sci. Robot.* **4**, eaau5872 (2019).
16. T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, Learning robust perceptive locomotion for quadrupedal robots in the wild. *Sci. Robot.* **7**, eabk2822 (2022).
17. T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, S. Levine, Learning to walk via deep reinforcement learning, in *Proceedings of the Robotics: Science and Systems* (2019).

18. W. Yu, J. Tan, Y. Bai, E. Coumans, S. Ha, Learning fast adaptation with meta strategy optimization. *IEEE Robot. Autom. Lett.* **5**, 2950–2957 (2020).
19. X. B. Peng, M. Andrychowicz, W. Zaremba, P. Abbeel, Sim-to-real transfer of robotic control with dynamics randomization, in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation* (IEEE, 2018), pp. 3803–3810.
20. R. Kaushik, T. Anne, J. Mouret, Fast online adaptation in robotics through meta-learning embeddings of simulated priors, in *Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2020), pp. 5269–5276.
21. E. Coumans, Y. Bai, Pybullet, a python module for physics simulation for games, robotics and machine learning (2016);pybullet.org.
22. E. Todorov, T. Erez, Y. Tassa, MuJoCo: A physics engine for model-based control, in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2012), pp. 5026–5033.
23. J. Hwangbo, J. Lee, M. Hutter, Per-contact iteration method for solving contact dynamics. *IEEE Robot. Autom. Lett.* **3**, 895–902 (2018).
24. P. A. Cundall, O. D. L. Strack, A discrete numerical model for granular assemblies. *Geotechnique* **29**, 47–65 (1979).
25. T. Pöschel, T. Schwager, *Computational Granular Dynamics: Models and Algorithms* (Springer-Verlag, 2005).
26. C. Li, T. Zhang, D. I. Goldman, A terradynamics of legged locomotion on granular media. *Science* **339**, 1408–1412 (2013).
27. J. Aguilar, D. I. Goldman, Robophysical study of jumping dynamics on granular media. *Nat. Phys.* **12**, 278–283 (2016).
28. S. Agarwal, A. Karsai, D. I. Goldman, K. Kamrin, Surprising simplicity in the modeling of dynamic granular intrusion. *Sci. Adv.* **7**, eabe0631 (2021).
29. M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, arXiv:1703.01365 (2017).
30. J. W. Glasheen, T. A. McMahon, A hydrodynamic model of locomotion in the Basilisk lizard. *Nature* **380**, 340–342 (1996).
31. H. Katsuragi, D. Durian, Drag force scaling for penetration into granular media. *Phys. Rev. E* **87**, 052208 (2013).
32. S. Hochreiter, J. Schmidhuber, Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
33. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv:1707.06347 (2017).
34. M. Andrychowicz, B. Baker, M. Chociej, R. J. Zefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, W. Zaremba, Learning dexterous in-hand manipulation. *Int. J. Robot. Res.* **39**, 3–20 (2020).
35. G. An, The effects of adding noise during backpropagation training on a generalization performance. *Neural Comput.* **8**, 643–674 (1996).
36. T. Fields, G. Hsieh, J. Chenou, Mitigating drift in time series data with noise augmentation, in *Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence* (IEEE, 2019), pp. 227–230.
37. B. Katz, J. D. Carlo, S. Kim, Mini cheetah: A platform for pushing the limits of dynamic quadruped control, in *Proceedings of the 2019 IEEE International Conference on Robotics and Automation* (IEEE, 2019), pp. 6295–6301.
38. S. Bai, J. Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271 [cs.LG] (4 Mar 2018).
39. H. Katsuragi, D. Durian, Unified force law for granular impact cratering. *Nat. Phys.* **3**, 420–423 (2007).



**Acknowledgments**

**Funding:** This work was supported by the Samsung Research Funding & Incubation Center of Samsung Electronics under Project Number SRFC-IT2002-02. **Author contributions:** S.C. conceived the main idea of the simulation and training methods, set up the simulation, and trained control policies. G.J. designed the initial setup for training. S.C., J.P., and J.H. set up the hardware for experiments. S.C. and J.P. performed indoor experiments together. S.C., G.J., J.P., H.K., J.M., and J.H.L. conducted outdoor experiments. S.C. and J.H. analyzed the data. All authors refined ideas and contributed in the experiment design. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper or the Supplementary Materials.

Submitted 2 August 2022
Accepted 21 December 2022
Published 25 January 2023
10.1126/scirobotics.ade2256

Science Robotics

# Learning quadrupedal locomotion on deformable terrain

Suyoung Choi, Gwanghyeon Ji, Jeongsoo Park, Hyeongjun Kim, Juhyeok Mun, Jeong Hyun Lee, and Jemin Hwangbo

*Sci. Robot.* **8** (74), eade2256. DOI: 10.1126/scirobotics.ade2256







*Science Robotics* (ISSN 2470-9476) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Robotics* is a registered trademark of AAAS.