ANIMAL ROBOTS

Learning robust perceptive locomotion for quadrupedal robots in the wild

Takahiro Miki[1]*, Joonho Lee[1], Jemin Hwangbo[2], Lorenz Wellhausen[1], Vladlen Koltun[3], Marco Hutter[1]

Legged robots that can operate autonomously in remote and hazardous environments will greatly increase opportunities for exploration into underexplored areas. Exteroceptive perception is crucial for fast and energy-efficient locomotion: Perceiving the terrain before making contact with it enables planning and adaptation of the gait ahead of time to maintain speed and stability. However, using exteroceptive perception robustly for locomotion has remained a grand challenge in robotics. Snow, vegetation, and water visually appear as obstacles on which the robot cannot step or are missing altogether due to high reflectance. In addition, depth perception can degrade due to difficult lighting, dust, fog, reflective or transparent surfaces, sensor occlusion, and more. For this reason, the most robust and general solutions to legged locomotion to date rely solely on proprioception. This severely limits locomotion speed because the robot has to physically feel out the terrain before adapting its gait accordingly. Here, we present a robust and general solution to integrating exteroceptive and proprioceptive perception for legged locomotion. We leverage an attention-based recurrent encoder that integrates proprioceptive and exteroceptive input. The encoder is trained end to end and learns to seamlessly combine the different perception modalities without resorting to heuristics. The result is a legged locomotion controller with high robustness and speed. The controller was tested in a variety of challenging natural and urban environments over multiple seasons and completed an hour-long hike in the Alps in the time recommended for human hikers.

INTRODUCTION

Legged robots can carry out missions in challenging environments that are too far or too dangerous for humans, such as hazardous areas and the surfaces of other planets. Legs can walk over challenging terrain with steep slopes, steps, and gaps that may impede wheeled or tracked vehicles of similar size. There has been notable progress in legged robotics (*1*–*5*), and several commercial platforms are being deployed in the real world (*6*–*10*).

However, until now, legged robots could not match the performance of animals in traversing challenging real-world terrain. Many legged animals, such as humans and dogs, can briskly walk or run in such environments by foreseeing the upcoming terrain and planning their footsteps based on visual information (*11*). Animals naturally combine proprioception and exteroception to adapt to highly irregular terrain shape and surface properties such as slipperiness or softness, even when visual perception is limited. Endowing legged robots with this ability is a grand challenge in robotics.

One of the biggest difficulties lies in reliable interpretation of incomplete and noisy perception for control. Exteroceptive information provided by onboard sensors is incomplete and often unreliable in real-world environments. Stereo camera–based depth sensors, which most existing legged robots rely on (*6*, *9*, *12*), require texture to perform stereo matching and consequently struggle with low-texture surfaces or when parts of the image are under- or overexposed. Time-of-flight (ToF) cameras often fail to perceive dark surfaces and become noisy under sunlight (*13*). In general, sensors that rely on light to infer distance are prone to producing artifacts on highly reflective surfaces, because the sensors assume that light travels in a straight path. In addition, depth sensors by nature cannot distinguish soft unstable surfaces, such as vegetation, from rigid ones. An elevation map is commonly used to represent geometric terrain information extracted from depth sensor measurements (*14*–*17*). It relies on the robot's estimated pose and is therefore affected by errors in this estimate. Other common sources of uncertainty in the map are occlusion or temporal inconsistency of the measurements due to dynamic objects. Most existing methods that rely on onboard terrain perception are still vulnerable to these failures.

Conventional approaches assume that the terrain information and any uncertainties encoded in the map are reasonably accurate, and the focus shifts solely to generating the motion. Offline methods use a prescanned terrain map, compute a handcrafted cost function over the map, and optimize a trajectory that is replayed on the robot (*18*, *19*). They assume perfect knowledge of the full terrain and robot states and plan complex motions with long planning times. Online methods generally use a similar approach but use only onboard resources to construct a map and continuously replan trajectories during execution (*20*–*24*). Recently, faster locomotion has been achieved by reducing the planning time with heuristics (*25*–*27*) or using convolutional neural networks to calculate foothold cost more efficiently (*27*). Recently, a bipedal robot, Atlas, demonstrated parkour over complex obstacles (*28*). It leverages preplanned motion reference and optimizes its motion online by using onboard LiDAR sensor data. Overall, the focus of all the approaches mentioned above is on picking footholds and generating trajectories given accurate terrain information. Some works (*14*, *17*) represent the statistical uncertainty of the measurements in the map, but its use is limited to heuristically defined foot placement rules to avoid risky areas (*24*). Such methods can only handle explicitly modeled uncertainties and are not robust to the variety of perception failures encountered in the wild.

Data-driven methods have recently been introduced to incorporate more complex dynamics without compromising real-time performance.

[1]Robotic Systems Lab, ETH-Zürich, Zürich, Switzerland. [2]Robotics & Artificial Intelligence Lab, KAIST, Daejeon, Korea. [3]Intelligent Systems Lab, Intel, Jackson, WY, USA.
*Corresponding author. Email: tamiki@ethz.ch

Learning-based quadrupedal or bipedal locomotion for simulated characters has been achieved by using reinforcement learning (RL) (*29–32*), and realistic robot models were used in recent works (*33*). However, these works were only conducted in simulation. Recently, RL-based locomotion controllers have been successfully transferred to physical robots (*3*, *4*, *34–40*). Hwangbo *et al.* (*3*, *41*) realized quadrupedal locomotion and recovery on flat ground with a physical robot by using learned actuator dynamics to facilitate simulation-to-reality (sim-to-real) transfer. Lee *et al.* (*4*) extended this approach and enabled rough-terrain locomotion by simulating challenging terrain in a privileged training setup with an adaptive curriculum. Peng *et al.* (*35*) used imitation learning to transfer animal motion to a legged robot. However, these methods do not use any visual information.

To add exteroceptive information to locomotion learning, Gangapurwala *et al.* (*42*) combined a learning-based foothold planner and a model-based whole-body motion controller to transfer policies to the real world in a laboratory setting. Their applications are limited to rigid terrain with mostly flat surfaces and are still constrained in their deployment range. Their performance is tightly bound to the quality of the map, which often becomes unreliable in the field.

In both model-based and learning-based approaches, the assumption of flawless map quality precludes the application of these methods in uncontrolled outdoor environments. Handling uncertainties in terrain perception remains an open problem. Existing controllers avoid catastrophic failures by simply refraining from using visual information in outdoor environments (*2*, *4*, *38*) or by adding heuristically defined reflex rules (*43*, *44*).

Here, we present a terrain-aware locomotion controller for quadrupedal robots that overcomes the limitations of previous approaches and enables robust traversal of harsh natural terrain at unprecedented speeds (Movie 1). At its core, the controller is based on a principled solution to incorporating exteroceptive perception into locomotion control.

The key component is a recurrent encoder that combines proprioception and exteroception into an integrated belief state. The encoder is trained in simulation to capture ground-truth information about the terrain given exteroceptive observations that may be incomplete, biased, and noisy. The belief state encoder is trained end to end to integrate proprioceptive and exteroceptive data without resorting to heuristics. It learns to take advantage of the foresight afforded by exteroception to plan footholds and accelerate locomotion when exteroception is reliable and can seamlessly fall back to robust proprioceptive locomotion when needed. The learned controller thus combines the best of both worlds: the speed and efficiency afforded by exteroception and the robustness of proprioception.

The controller is trained via privileged learning (*45*). We first train a teacher policy via RL with full access to privileged information in the form of the ground-truth state of the environment. This privileged training enables the teacher policy to discover the optimal behavior given perfect knowledge of the terrain. We then train a student policy that only has access to information that is available in the field on the physical robot. The student policy is built around our belief state encoder and trained via imitation learning. The student policy learns to predict the teacher's optimal action given only partial and noisy observations of the environment.

Once the student policy is trained, we deploy it on the robot without any fine-tuning. The controller gets onboard sensor observations and a desired velocity command and outputs each joint's target position as the action. The robot perceives the environment by leveraging a robot-centric elevation map. The elevation map serves as an abstraction layer between sensors and the locomotion controller, making our method independent of depth sensor choices. It works with no fine-tuning with different sensors, such as stereo cameras or LiDAR. Because the policy was trained to handle large noises, bias, and gaps in the elevation map, the robot can continue walking even when mapping fails or the sensors are physically broken.

The presented approach achieves substantial improvements over the state of the art (*4*) in locomotion speed and obstacle traversability while maintaining exceptional robustness. Our key contribution is a method for combining multimodal perception and demonstrating with extensive hardware experiments that the resulting control policy is robust against various exteroceptive failures. Handling exteroception failures has been a challenging problem in robotics. Our approach constitutes a general framework for robust deployment of complex autonomous machines in the wild.

RESULTS

Fast and robust locomotion in the wild

We deployed our controller in a wide variety of terrain, as shown in Fig. 1 and Movie 1. This includes alpine, forest, underground, and urban environments. The controller was consistently robust and had zero falls during all deployments. Because of the exteroceptive perception, the robot could anticipate the terrain and adapt its motion to achieve fast and smooth walking. This was particularly notable for structures that require high foot clearance, such as stairs and large obstacles. The robot was able to leverage exteroceptive input to conquer terrain that was beyond the capabilities of prior work that did not use exteroception (*4*).

ANYmal successfully traversed challenging natural environments with steep inclination, slippery surfaces, grass, and snow (Fig. 1, A to J). The robot was robust under these conditions, even when occlusion and surface properties such as high reflectance impeded exteroception. Our controller was also robustly deployed in underground environments with loose gravel, sand, dust, water, and limited illumination (Fig. 1, K to N).

Urban environments also present important challenges (Fig. 1, O to R). For traversing stairs, the state-of-the-art quadrupedal robot Spot from Boston Dynamics requires that a dedicated mode is engaged, and the robot must be properly oriented with respect to the stairs [(*44*), p. ~33]. In contrast, our controller does not require any special mode for stairs and can traverse stairs natively in any direction and any orientation, such as sideways, diagonally, and turning around on the stairway. See movie S1 for demonstrations of smooth and robust stair traversal in arbitrary directions with our controller.

The controller was also robust to combinations of different challenges, as can be seen with snow on stairs in Fig. 1R. Snow makes stairs slippery and yields incomplete and erroneous exteroceptive data. Depth sensors either fail due to the high reflectivity of snow or estimate the surface profile to be on top of the snow, whereas the robot's legs sink below this level. Foot slippage in snow can also cause large drift in the kinematic pose estimation (*46*), making the map even more inconsistent. Nevertheless, the controller remained consistently robust, with zero failures in this regime as well.

A hike in the Alps

To further evaluate the robustness of our controller, we conducted a hiking experiment in which we tested whether ANYmal could

Fig. 1. Robust locomotion in the wild. The presented locomotion controller was extensively tested in a variety of complex environments such as natural [(**A**) to (**J**)], underground [(**K**) to (**N**)] or various stairs [(**O**) to (**R**)] over multiple seasons. The controller overcame a whole spectrum of real-world challenges, often encountering them in combination. These include slippery surfaces [(M) and (R)], steep inclinations [(C) to (E)], complex terrain, and vegetation in natural environments [(B), (C), (F), and (I)]. In search-and-rescue scenarios, the controller dealt with steep stairs [(F), (G), and (O) to (R)], unknown payloads (I), and perception-degrading fog (P). Reflective surfaces (N), loose ground [(K) and (M)], low light, and water puddles were encountered in underground cave systems [(K) to (N)]. Soft and slippery snow piled up in the winter [(J) and (R)]. The controller traversed these environments with zero failures.

complete an hour-long hiking loop on the Etzel mountain in Switzerland. The hiking route was 2.2 km long, with an elevation gain of 120 m. Completing the trail required traversing steep inclinations, high steps, rocky surfaces, slippery ground, and tree roots (Fig. 2). As seen in Movie 2, ANYmal completed the entire hike without any failure, stopping only to fix a detached shoe and swap batteries. The robot was able to reach the summit in 31 min, which is faster than the expected human hiking duration indicated in the official signage (35 min as shown in Fig. 2), and finished the entire path in 78 min, virtually the same duration suggested by a hiking planner (76 min), which rates the hike "difficult" (*47*). The difficulty levels are chosen from "easy," "moderate," and "difficult," calculated by

Movie 1. Wild ANYmal: Robust zero-shot perceptive locomotion.

combining the required fitness level, sport type, and the technical complexity (*48*).

During the hike, the controller faced various challenges. The ascending path reached inclinations of up to 38% with rocky and wet surfaces (Fig. 2, B and C). On the descent through a forest, tree roots formed intricate obstacles, and the ground proved very slippery (Fig. 2, G and H).

Vegetation above the robot sometimes introduced severe artifacts into the estimated elevation map. Despite all the challenges, the robot finished the hike without any human help and without a single fall.

Exteroceptive challenges

In this section, we examine how the terrain was perceived by the robot under conditions that are challenging for exteroception. The robot perceives the environment in the form of height samples from an elevation map constructed from point-cloud input, as seen in Fig. 3A. We used LiDAR in some experiments (Fig. 3, D to G) and active stereo cameras in others (Fig. 3, B and C) to test the robustness of the controller to the sensing modality.

We encountered many circumstances in which exteroception provides incomplete or misleading input. As shown in Fig. 3 (B to G), the estimated elevation map can be unreliable due to sensing failures, limitations of the 2.5D height map representation, or viewpoint restrictions due to onboard sensing.

Because most depth sensors rely on light to infer distance, either through ToF measurements or stereo disparity, they commonly struggle with reflective or translucent surfaces. Figure 3B shows such a sensing failure, where the reflective metal floor induced large depth outliers that appear as a trench in the elevation map. Figure 3C shows a sensing failure in the presence of snow. Because snow is highly reflective and has very little texture, stereo cameras could not infer depth, which led to an empty map.

The 2.5D elevation map representation cannot accurately represent overhanging objects such as tree branches or low ceilings (*17*). These were integrated into the height field and were misrepresented as tall obstacles (Fig. 3D). In addition, because the map cannot distinguish between rigid and soft materials, the map gave misleading information in soft vegetation or deep snow (Fig. 3E).

Slippery or deformable surfaces caused odometry drift because they violate the assumption of stable footholds commonly adopted by kinematic pose estimators (*46*). Because map construction relies on such pose estimation to register consecutive input point clouds, the map became inaccurate in such circumstances (Fig. 3F). Furthermore, because the sensors were only located on the robot itself, areas behind structures were occluded and not presented in the map, which was especially problematic during uphill walking (Fig. 3G).

Overall, our controller could handle all of these challenging conditions gracefully, without a single failure. The belief state estimator was trained to assess the reliability of exteroceptive information and made use of it to the extent possible. When exteroceptive information was incomplete, noisy, or misleading, the controller could always gracefully degrade to proprioceptive locomotion, which was shown to be robust (*4*). The controller thus aims to achieve the best of both worlds: achieving fast predictive locomotion when exteroceptive information is informative but seamlessly retaining the robustness of proprioceptive control when it is not.

Evaluating the contribution of exteroception

We conducted controlled experiments to quantitatively evaluate the contribution of exteroception. We compared our controller with a proprioceptive baseline (*4*) that does not use exteroception.

First, we compared the success rate of overcoming fixed-height steps as shown in Fig. 4A. Wooden steps of various heights (from 12 to 36.5 cm) were placed ahead of the robot, which performed 10 trials to overcome each step with a fixed velocity command. A trial was considered successful if the robot overcame the step within 5 s.

The success rate of the proprioceptive baseline dropped at a 20-cm step height, when the front legs started frequently getting stuck at the step (Fig. 4B). Even when the front legs successfully overcame the step, the hind legs often failed to fully step up. In contrast, our controller reliably traversed steps of up to 30.5 cm in height. Because our controller could anticipate the step, it lifted its legs higher without making physical contact first and leaned its body forward to let the hind leg swing over the step (Fig. 4A). Until this height, the dominant failure reason was the robot evading the step sideways instead of falling. When approaching steps higher than 32 cm, our controller hesitated to walk forward because it learned that steps of such height are at or above the robot's physical limits and are likely to incur a high cost.

We also tested the two controllers in an obstacle course, as shown in Fig. 4 (C and D). In this experiment, the robot was given a fixed path over the obstacles and tracked it using a pure pursuit controller (*49*). The path traverses several types of obstacles: an inclined platform, a raised platform, stairs, and a pile of blocks. The platforms are 20 cm high, the stairs are 17 cm high and 29 cm deep each, and the blocks are each 20 cm in both height and depth. Our controller followed the given path smoothly without any assistance, as shown in Fig. 4C. The exteroceptive perception provided advance information on the upcoming obstacles, allowing the controller to adjust the robot's motion before it made contact with the obstacles, facilitating fast and smooth motion through the obstacle course. The baseline, on the other hand, failed to track the path without human assistance. During execution, it got stuck on all three obstacles, and we had to lift and push the robot to continue the experiment (Fig. 4D).

In addition, we measured the maximum locomotion speed of both controllers over flat ground and in the presence of obstacles. Figure 4E shows the experimental setup. We gave the controller a constant forward, lateral, or turning command and recorded the velocity on flat ground and over a 20-cm step. Note that the baseline controller only receives a directional command and learns to walk as fast as possible in the commanded direction (*4*). Our controller

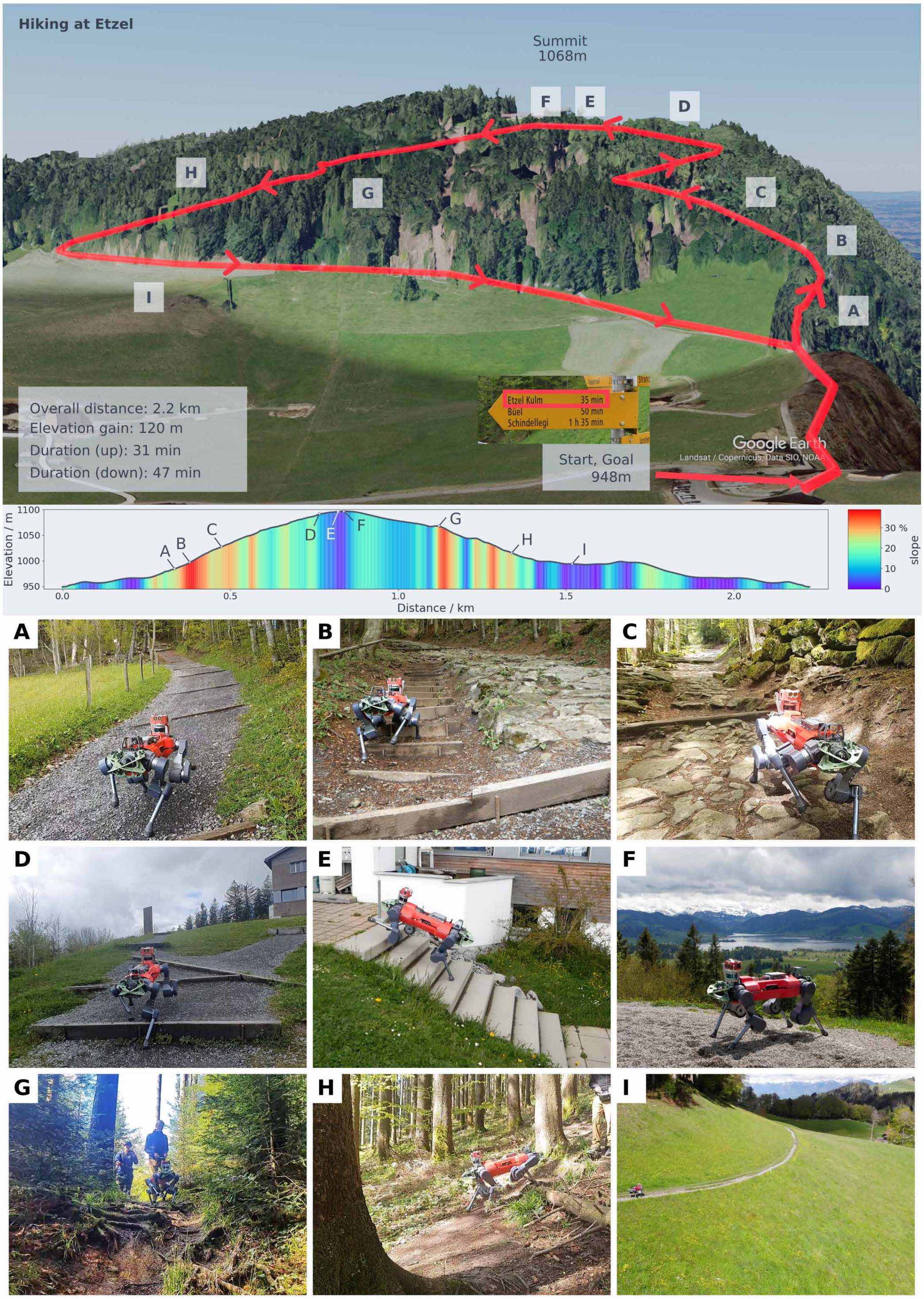

Fig. 2. A hike on the Etzel mountain in Switzerland, completed by ANYmal with our locomotion controller. The 2.2-km route—with 120 m of elevation gain and inclinations up to 38%—encompasses a variety of challenging terrains [(**A**) to (**I**)]. ANYmal reached the summit faster than the human time indicated in the official signage and finished the entire route in virtually the same time as given by a hiking guide (*47*).

walked at 1.2 m/s, whereas the baseline could only achieve 0.6 m/s on flat ground in both the forward and lateral directions. The difference became even more pronounced over the obstacle. Our controller could traverse the obstacle without any notable slowdown, whereas the baseline was stymied. The turning velocity showed the biggest difference between the baseline policy and ours. Our controller could turn at 3 rad/s, but the baseline policy could only turn at 0.6 rad/s: a fivefold difference.

These results show gains by our controller over the proprioceptive baseline. Exteroception enabled our controller to traverse challenging environments more successfully and at higher speeds in comparison with pure proprioception. Further quantitative performance evaluation is provided in section S2.

Evaluating robustness with belief state visualization

To examine how our controller integrates proprioception and exteroception, we conducted a number of controlled experiments. We tested with two types of obstacles that provide ambiguous or misleading exteroceptive input: an opaque foam obstacle that appears solid but cannot support a foothold and a solid but transparent obstacle. We placed each obstacle ahead of the robot and commanded the robot to walk forward at a constant velocity.

The sensors perceived the foam block as solid, and the robot consequently prepared to step on it but could not achieve a stable foothold due to the deformation of the foam. Figure 5A shows how the internal belief state (blue) was revised as the robot encounters the misleading obstacle: The controller initially trusted the exteroceptive input (red) but quickly revised its estimate of terrain height upon contact. Once the correct belief had been formed, it was retained even after the foot left the ground, showing that the controller retains past information due to its recurrent structure.

The transparent obstacle is a block made of clear, acrylic plates that were not accurately perceived by the onboard sensors (Fig. 5B).

The robot therefore walked as if it was on flat ground until it made contact with the step, at which point it revised its estimate of terrain profile upward and changed its gait accordingly.

Movie 2. Hiking at Etzel.

In the next experiment, we simulated complete exteroception failure by physically covering the sensors, thus making them fully uninformative (Fig. 5, C and D). The robot was commanded to walk up and down two steps of stairs. With an unobstructed sensor, the controller traversed the stairs gracefully, without any unintended contact with the stair risers, adjusting its footholds and body posture to step down the stairs softly. When the sensors were covered, the map had no information, and the controller received random noise as input. Under this condition, the robot made contact with the riser of the first stair, which could not be perceived in advance, revised its estimate of the terrain profile, adjusted its gait accordingly, and successfully climbed the stairs. On the way down, the blinded robot made a hard landing with its front feet but kept its balance and stepped down softly with its hind legs.

Last, we tested locomotion over an elevated slippery surface (Fig. 5E). After the robot stepped onto the slippery platform, it detected the low friction and adapted its behavior to step faster and keep its balance. The momentarily sliding feet violated the assumption of

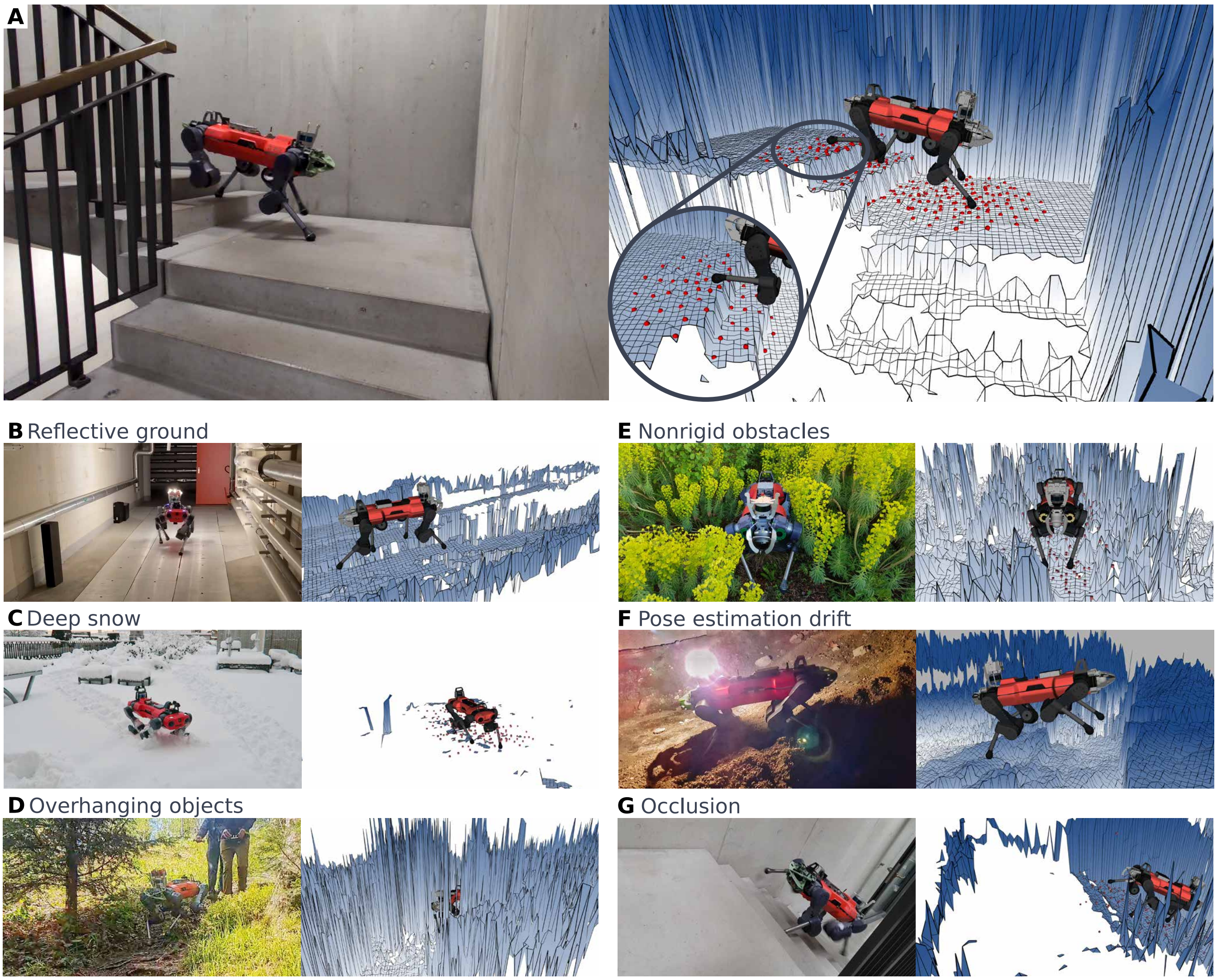

Fig. 3. Exteroceptive representation and challenges. Our locomotion controller perceives the environment through height samples (red dots) from an elevation map (**A**). The controller is robust to many perception challenges commonly encountered in the field: missing map information due to sensing failure (**B**, **C**, and **G**) and misleading map information due to nonrigid terrain (**D** and **E**) and pose estimation drift (**F**).

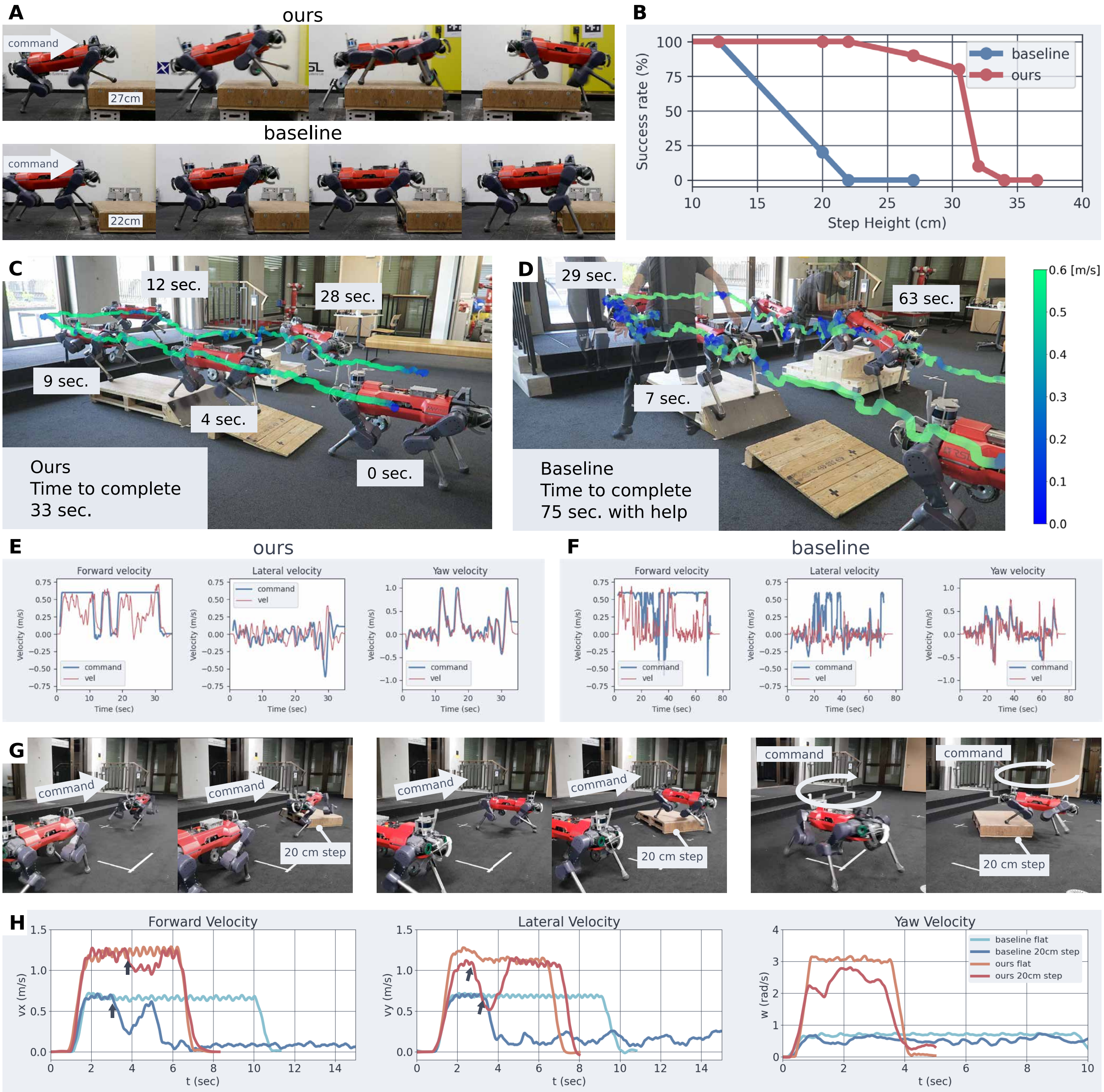

Fig. 4. We compared the presented controller with a proprioceptive baseline. An experiment with steps of varying height shows that our controller can overcome notably higher obstacles than the baseline (**A** and **B**). Our method completes an obstacle course in less than half the time of the baseline and without requiring any human help (**C** and **D**). As seen in the graphs, our controller could follow the command more precisely. Note that the directional command plotted in (F) is scaled to 0.6 m/s. (**E** and **F**) Our controller can maintain double the linear velocity of the baseline and achieves a fivefold increase in turning speed. The arrows indicate when the robot reached the step (**G** and **H**).

the kinematic pose estimator, which, in turn, destabilized the estimated elevation map and rendered exteroception uninformative during this time. The controller seamlessly fell back on proprioception until the estimated elevation map stabilized and exteroception became informative again.

DISCUSSION

We have presented a fast and robust quadrupedal locomotion controller for challenging terrain. The controller seamlessly integrates exteroceptive and proprioceptive input. Exteroceptive perception enables the robot to traverse the environment quickly and gracefully

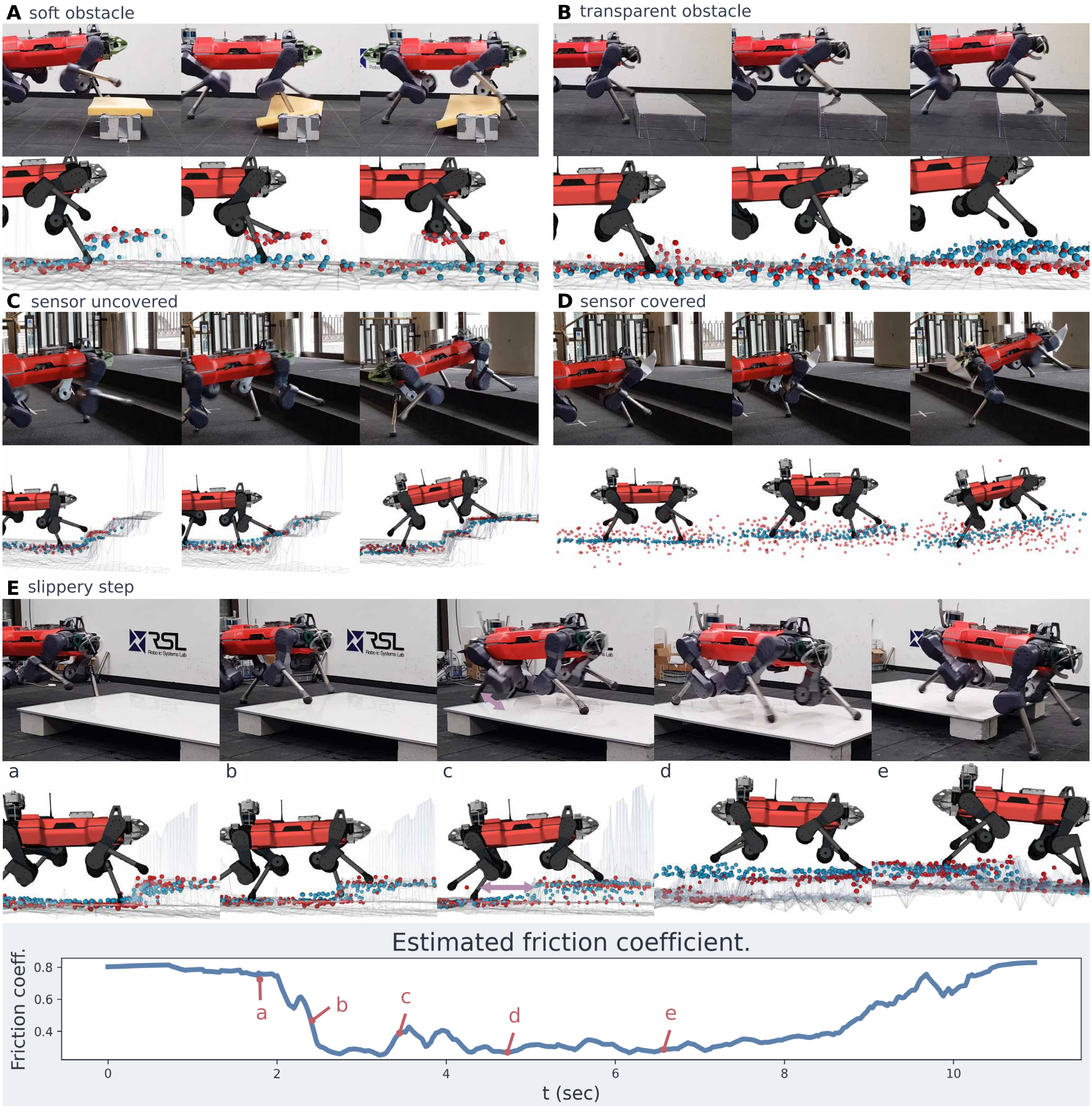

Fig. 5. Internal belief state inspection during perceptive failure using a learned belief decoder. Red dots indicate height samples given as input to the policy. Blue dots show the controller's internal estimate of the terrain profile. (**A**) After stepping on a soft obstacle that cannot support a foothold, the policy correctly revises its estimate of the terrain profile downward. (**B**) A transparent obstacle is correctly incorporated into the terrain profile after contact is made. (**C**) With operational sensors, the robot swiftly and gracefully climbs the stairs, with no spurious contacts. (**D**) When the robot is blinded by covering the sensors, the policy can no longer anticipate the terrain but remains robust and successfully traverses the stairs. (**E**) When stepping onto a slippery platform, the policy identifies low friction and compensates for the induced pose estimation drift. The graph shows a decoded friction coefficient.

by anticipating the terrain and adapting its gait accordingly before contact is made. When exteroceptive perception is misleading, incomplete, or missing altogether, the controller smoothly transitions to proprioceptive locomotion. The controller remains robust under all conditions, including when the robot is effectively blind. The integration of exteroceptive and proprioceptive inputs is learned end to end and does not require any hand-coded rules or heuristics. The result is a rough-terrain legged locomotion controller that combines the speed and grace of vision-based locomotion with the high robustness of proprioception.

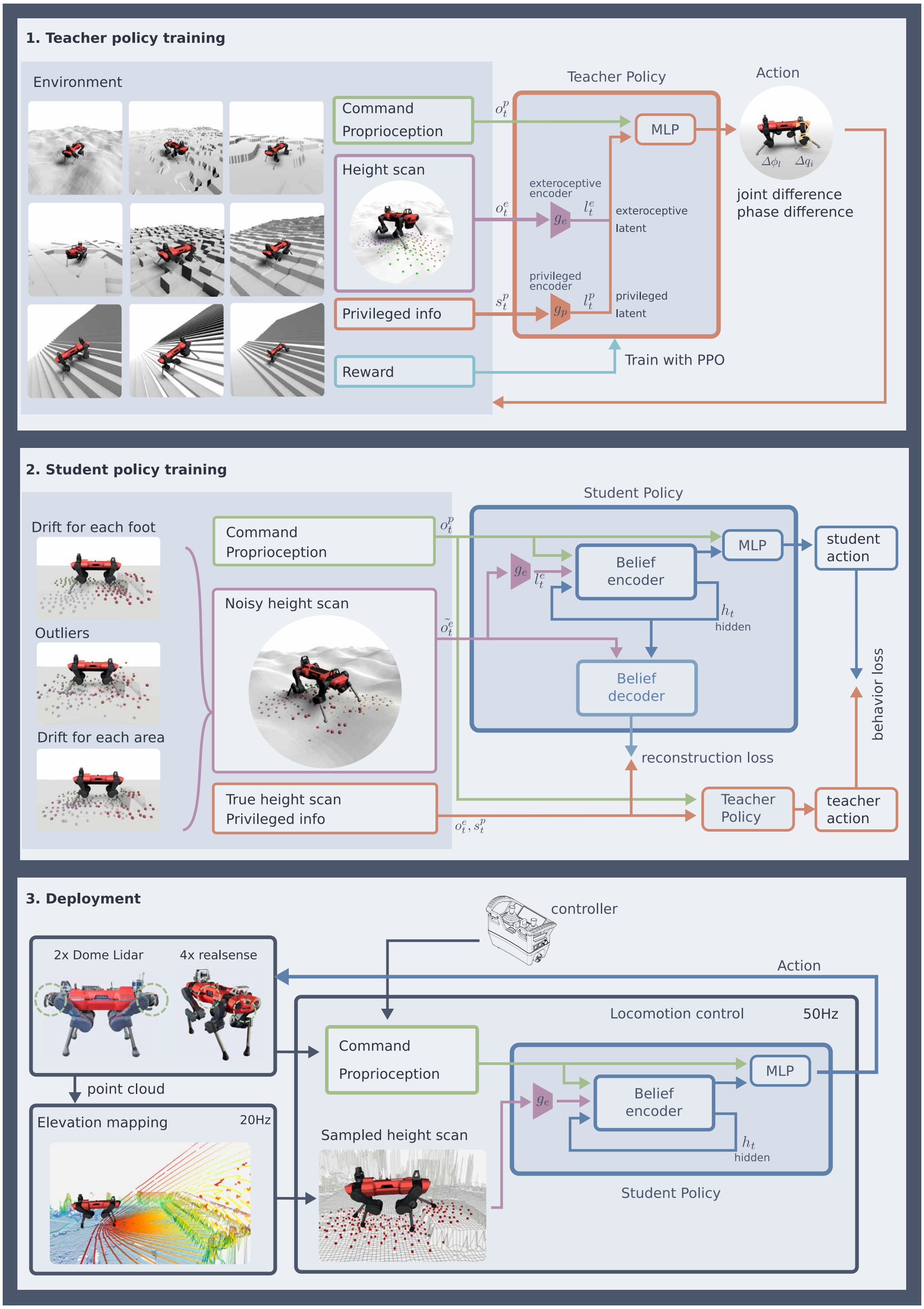

Fig. 6. Overview of the training methods and deployment. We first train a teacher policy with access to privileged simulation data using RL. This teacher policy is then distilled into a student policy, which is trained to imitate the teacher's actions and to reconstruct the ground-truth environment state from noisy observations. We deploy the student policy zero-shot on real hardware using height samples from a robot-centric elevation map.

This combination of speed and high robustness has been validated through controlled experiments and extensive deployments in the wild, including an hour-long hiking route in the Alps that is rated "difficult" (*47*). The entire route was completed by the robot without human assistance (other than reattaching a detached shoe and swapping the batteries) in the recommended time for completion of this route by human hikers.

Our work expands the operational domain of legged robots and opens up previously unexplored frontiers in autonomous navigation. Navigation planners no longer need to identify ground type or to switch modes during autonomous operation. Our controller was used as the default controller in the Defense Advanced Research Projects Agency Subterranean Challenge missions of team Cerberus (*50*, *51*), which has won the first prize in the finals (*52*). In this challenge, our controller drove ANYmals to operate autonomously over extended periods of time in underground environments with rough terrain, obstructions, and degraded sensing in the presence of dust, fog, water, and smoke (*53*). Our controller played a crucial role because it enabled four ANYmals to explore over 1700 m in all three types of courses—tunnel, urban, and cave—without a single fall.

Possible extensions

Future work could explicitly use the uncertainty information in the belief state. Currently, the policy uses uncertainty only implicitly to estimate the terrain. For example, in front of a narrow cliff or a stepping stone, the elevation map does not provide sufficient information due to occlusion. Therefore, the policy assumes a continuous surface, and, as a result, the robot might step off and fall. Explicitly estimating uncertainty may allow the policy to become more careful when exteroceptive input is unreliable, for example, using the robot's foot to probe the ground if the policy is unsure about it. In addition, our current implementation obtains perceptual information through an intermediate state in the form of an elevation map, rather than directly ingesting raw sensor data. This has the advantage that the

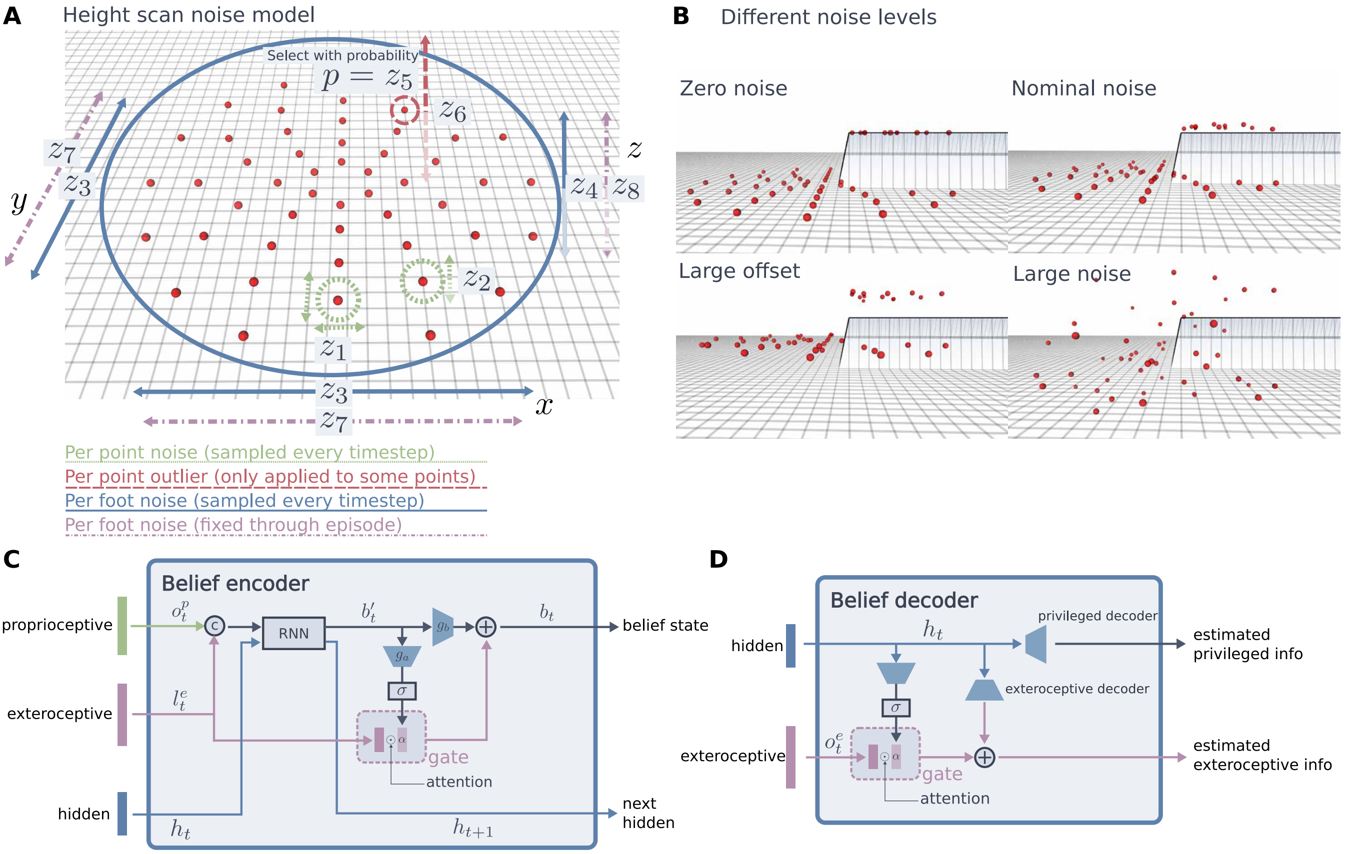

Fig. 7. Details of robust terrain perception components. (**A**) During student training, random noise is added to the height samples. The noise is sampled from a Gaussian distribution $\mathcal{N}(0, z^l \in \mathbb{R}^8)$, where each z_i^l controls a different noise component i per leg l. (**B**) We use multiple noise configurations z to simulate different operating conditions. "Zero noise" is applied during teacher training, whereas "nominal noise" represents normal mapping conditions during student training. "Large offset" noise simulates large map offsets due to pose estimation drift or deformable terrain surfaces. "Large noise" simulates a complete lack of terrain information due to occlusion or sensor failure. (**C**) The student policy belief encoder incorporates a recurrent core and an attentional gate that integrates the proprioceptive and exteroceptive modalities. The gate explicitly controls which aspects of exteroceptive data should pass through. (**D**) The belief decoder has a gate for reconstructing the exteroceptive data. It is only used during training and for introspection into the belief state.

model is independent of the specific exteroceptive sensors. (We use LiDAR and stereo cameras in different deployments, with no retraining or fine-tuning.) However, the elevation map representation omits detail that may be present in the raw sensory input and may provide additional information concerning material and texture. Furthermore, our elevation map construction relies on a classical pose estimation module that is not trained jointly with the rest of the system. Appropriately folding the processing of raw sensory input into the network may further enhance the speed and robustness of the controller. In addition, an occlusion model could be learned, such that the policy understands that there is an occlusion behind the cliff and avoids stepping off it. Another limitation is the inability to complete locomotion tasks, which would require maneuvers very different from normal walking, for example, recovering from a leg stuck in narrow holes or climbing onto high ledges.

MATERIALS AND METHODS

Overview

We train a neural network policy in simulation and then perform zero-shot sim-to-real transfer. Our method consists of three stages, as illustrated in Fig. 6.

First, a teacher policy is trained with RL to follow a random target velocity over randomly generated terrain with random disturbances. The policy has access to privileged information such as noiseless terrain measurements, ground friction, and the disturbances that were introduced.

In the second stage, a student policy is trained to reproduce the teacher policy's actions without using this privileged information. The student policy constructs a belief state to capture unobserved information using a recurrent encoder and outputs an action based on this belief state. During training, we leverage two losses: a behavior cloning loss and a reconstruction loss. The behavior cloning loss aims to imitate the teacher policy. The reconstruction loss encourages the encoder to produce an informative internal representation.

Last, we transfer the learned student policy to the physical robot and deploy it in the real world with onboard sensors. The robot constructs an elevation map by integrating depth data from onboard sensors and samples height readings from the constructed elevation map to form the exteroceptive input to the policy. This exteroceptive input is combined with proprioceptive sensory data and is given to the neural network, which produces actuator commands.

Problem formulation

We formulate our control problem in discrete time dynamics, where the environment is fully defined by the state s_t at time step t. The policy performs an action a_t and observes the environment via o_t, which comes from an observation model $\mathcal{O}(o_t \mid s_t, a_t)$. Then, the environment moves to the next state s_{t+1} with transition probability $P(s_{t+1} \mid s_t, a_t)$ and returns a reward r_{t+1}.

When all states are observable such that $o_t = s_t$, this can be considered as Markov decision process (MDP). When there is unobservable information, however, such as external forces or full terrain information in our case, the dynamics are modeled as a partially observable Markov decision process (POMDP).

The RL objective is to find a policy π^* that maximizes the expected discounted reward over the future trajectory, such that

$$\pi^* = \underset{a}{\operatorname{argmax}} \mathrm{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$$

A number of RL algorithms have been developed to solve fully observable MDPs and are readily available to be used for training. However, the case of POMDPs is more challenging because the state is not fully observable. This is often overcome by constructing a belief state b_t from a history of observations $\{o_0, \cdots, o_t\}$ in an attempt to capture the full state. In deep RL, this is frequently done by stacking a sequence of previous observations (*54*) or by using architectures that can compress past information, such as recurrent neural network (RNN) (*55*, *56*) or temporal convolutional network (*4*, *57*).

Training a complex neural network policy that handles sequential data naively from scratch can be time-consuming (*4*). Therefore, we use privileged learning (*45*), in which we first train a teacher policy with privileged information and then distill the teacher policy into a student policy via supervised learning.

Training environment

We use RaiSim (*58*) as our simulator to build the training environment. There, we simulate multiple ANYmal-C robots on randomly generated rough terrain in parallel with an integrated actuator model (*3*) to close the reality gap.

Terrain

We define parameterized terrain as shown in Fig. 6 (1). The terrain is modeled as a height map; further details are provided in section S4.

In addition to terrains composed of a variety of slopes and steps, we modeled four different types of stairs in the training environment: standard, open, ledged, and random. We use boxes to form the stairs, because stair risers modeled by a height map are not perfectly vertical; we observed that the policy exploited these nonvertical edges in simulation, resulting in poor sim-to-real transfer.

Domain randomization

We randomize the masses of the robot's body and legs, the initial joint position and velocity, and the initial body orientation and velocity in each episode. In addition, external force and torque are applied to the body of the robot, and the friction coefficients of the feet are occasionally set to a low value to introduce slippage.

Termination

We terminate a training episode and start a new one when the robot reaches an undesirable state. Termination criteria are body collision with the ground, large body tilt, and exceeding the joint torque limit of the actuators. These criteria help shape the motion and obtain constraint-satisfying behaviors.

Teacher policy training

In the first stage of training, we aim to find an optimal reference control policy that has access to perfect, privileged information and enables ANYmal to follow a desired command velocity over randomly generated terrain. The desired command is generated randomly as a vector $\mathbf{v}_{\text{des}} \in \mathbb{R}^3 = (v_x, v_y, w)$, where v_x, v_y represents the longitudinal and lateral velocity, and w represents the yaw velocity, all in the robot's body frame.

We used proximal policy optimization (PPO) (*59*) to train the teacher policy. The teacher is modeled as a Gaussian policy, $a_t \sim \mathcal{N}(\pi_\theta(o_t = s_t), \sigma I)$, where π_θ is implemented by a multilayer perceptron (MLP) parameterized by θ, and σ represents the variance for each action.

Observation and action

The teacher observation is defined as $o_t^{\text{teacher}} = (o_t^p, o_t^e, s_t^p)$, where o_t^p refers to the proprioceptive observation, o_t^e refers to the exteroceptive observation, and s_t^p refers to the privileged state. o_t^p contains the body velocity, orientation, joint position and velocity history, action history, and each leg's phase. o_t^e is a vector of height samples around each foot with five different radii. The privileged state s_t^p includes contact states, contact forces, contact normals, friction coefficient, thigh and shank contact states, external forces and torques applied to the body, and swing phase duration.

Our action space is inspired by central pattern generators (*4*). Each leg $l = \{1, 2, 3, 4\}$ keeps a phase variable ϕ_l and defines a nominal trajectory based on the phase. The nominal trajectory is a stepping motion of the foot tip, and we calculate the nominal joint target $q_i(\phi_l)$ for each joint actuator $i = \{1, \cdots, 12\}$ using inverse kinematics. The action from the policy is the phase difference $\Delta\phi_l$ and the residual joint position target Δq_i. More details of the observation and action space are in section S5.

Policy architecture

We model the teacher policy π_θ as an MLP. It consists of three MLP components: exteroceptive encoder, privileged encoder, and the main network, as shown in Fig. 6. The exteroceptive encoder g_e receives o_t^e and outputs a smaller latent representation l_t^e

$$l_t^e = g_e(o_t^e)$$

The privileged encoder g_p receives the privileged state s_t^p and outputs a latent representation l_t^{priv}

$$l_t^{\text{priv}} = g_p(s_t^p)$$

These encoders compress each input to more compact representations and facilitate reuse of some of the teacher policy components by the student policy. More details on each layer are in section S6.

Rewards

We define a positive reward for following the command velocity and a negative reward for violating some imposed constraints. The command-following reward is defined as follows

$$r_{\text{command}} = \begin{cases} 1.0, & \text{if } \mathbf{v}_{\text{des}} \cdot \mathbf{v} > |\mathbf{v}_{\text{des}}| \\ \exp\left(-(\mathbf{v}_{\text{des}} \cdot \mathbf{v} - |\mathbf{v}_{\text{des}}|)^2\right), & \text{otherwise} \end{cases} \tag{1}$$

where $\mathbf{v}_{\text{des}} \in \mathbb{R}^2$ is the desired horizontal velocity, and $\mathbf{v} \in \mathbb{R}^2$ is the current horizontal body velocity with respect to the body frame. The same reward is applied to the yaw command as well. We penalize the velocity component orthogonal to the desired velocity as well as the body velocity around roll, pitch, and yaw. In addition, we use shaping rewards for body orientation, joint torque, joint velocity, joint acceleration, and foot slippage as well as shank and knee collision.

Body orientation reward was used to avoid strange postures of the body. Joint-related reward terms were used to avoid overly aggressive motion. Foot slippage and collision reward terms were used to avoid them. We tuned the reward terms by looking at the policy's behavior in simulation. In addition to the traversal performance, we checked the smoothness of the locomotion. All reward terms are specified in section S7.

Curriculum

We use two curricula to ramp up the difficulty as the policy's performance improves. One curriculum adjusts the terrain difficulty using an adaptive method (*4*), and the other changes elements such as reward or applied disturbances using a logistic function (*3*).

For the terrain curriculum, a particle filter updates the terrain parameters such that they remain challenging but achievable at any point during policy training (*4*). The second curriculum multiplies the magnitude of domain randomization and some reward terms (joint velocity, joint acceleration, orientation, slip, and thigh and shank contact) by a factor that is monotonically increasing and asymptotically trending to 1

$$c_{k+1} = (c_k)^d$$

where c_k is the curriculum factor at the kth iteration, and $0 < d < 1$ is the convergence rate.

Student policy training

After we train a teacher policy that can traverse various terrain with the help of privileged information, we distill it into a student policy that only has access to information that is available on the real robot. We use the same training environment as for the teacher policy but add additional noise to the student height sample observation: $o_t^{\text{student}} = (o_t^p, n(o_t^e))$, where $n(o_t^e)$ is a noise model applied to the height sample input. The noise model simulates different failure cases of exteroception frequently encountered during field deployment and is detailed below.

When there is a large noise in the exteroception, it becomes unobservable; thus, the dynamics is considered to be POMDP. In addition, the privileged states are not observable due to the lack of sensors to directly measure. Therefore, the policy needs to consider the sequential correlation to estimate the unobservable states. We propose to use a recurrent belief state encoder to combine sequences of both exteroception and proprioception to estimate the unobservable states as a belief state.

The student policy consists of a recurrent belief state encoder and an MLP, as shown in Fig. 6 (2). We denote the hidden state of the recurrent network by h_t. The belief state encoder takes o_t^{student} and h_t as input and outputs a latent vector b_t, which we refer to as the belief state. The goal is to match the belief state b_t with the feature vector $(l_t^e, l_t^{\text{priv}})$ of the teacher policy that encodes all locomotion-relevant information. We then pass o_t^p and b_t to the MLP, which computes the output action. The MLP structure remains the same as for the teacher policy, such that we can reuse the learned weights of the teacher policy to initialize the student network and speed up training.

Training is performed in a supervised fashion by minimizing two losses: a behavior cloning loss and a reconstruction loss. The behavior cloning loss is defined as the squared distance between the student action and the teacher action given the same state and command. The reconstruction loss is the squared distance between the noiseless height sample and privileged information (o_t^e, s_t^p) and their reconstruction from the belief state. We generate samples by rolling out the student policy to increase robustness (*60*, *61*).

Height sample randomization

During student training, we inject random noise into the height samples using a parameterized noise model $n(\tilde{o}_t^e \mid o_t^e, z)$, $z \in \mathbb{R}^{8 \times 4}$. We apply two different types of measurement noise when sampling the heights, as shown in Fig. 7A:

1) Shifting scan points laterally.

2) Perturbing the height values.

Each noise value is sampled from a Gaussian distribution, and the noise parameter z defines the variance. Both types of noise are applied in three different scopes, all with their own noise variance: per scan point, per foot, and per episode. The noise values per scan point and per foot are resampled at every time step, while the episodic noise remains constant for all scan points.

In addition, we define three mapping conditions with associated noise parameters z to simulate changing map quality and error sources, as shown in Fig. 7B.

1) Nominal noise assuming good map quality during regular operation.

2) Large offsets through high per-foot noise to simulate map offsets due to pose estimation drift or deformable terrain.

3) Large noise magnitude for each scan point to simulate a complete lack of terrain information due to occlusion or mapping failure.

These three mapping conditions are selected at the beginning of each training episode in a ratio of 60, 30, and 10%.

Last, we divide each training terrain into cells and add an additional offset to the height sample, depending on which cell it was sampled from. This simulates transitions between areas with different terrain characteristics, such as vegetation and deep snow. The parameter vector z is also part of a learning curriculum, and its magnitude increases linearly with training duration. The height sample representation is specified in more detail in section S8.

Belief state encoder

The recurrent belief state encoder encodes states that are not directly observable. To integrate proprioceptive and exteroceptive data, we introduce a gated encoder as shown in Fig. 7C, inspired by gated RNN models (*62*, *63*) and multimodal information fusion (*64*–*66*).

The encoder learns an adaptive gating factor that controls how much exteroceptive information should pass through. First, proprioception o_t^p, exteroceptive features from noisy observations $l_t^e = g_e(\tilde{o}_t^e)$, and hidden state s_t are encoded by the RNN module into the intermediate belief state $b_{t'}$. Then, the attention vector α is computed from $b_{t'}$. It controls how much exteroceptive information enters the final belief state b_t

$$b_{t'}, h_{t+1} = \mathrm{RNN}(o_t^p, l_t^e, h_t)$$

$$\alpha = \sigma(g_a(b_{t'}))$$

$$b_t = g_b(b_{t'}) + l_t^e \odot \alpha$$

Here, g_a and g_b are fully connected neural networks, and $\sigma(\cdot)$ is the sigmoid function.

The same gate is used in the decoder, where it is used to reconstruct the privileged information and the height samples (Fig. 7D). This is used to calculate a reconstruction loss that encourages the belief state to capture veridical information about the environment.

We use the GRU (*62*) as our RNN architecture. The evaluation of the effectiveness of the gate structure is presented in section S9.

Deployment

We deployed our controller on the ANYmal-C robot with two different sensor configurations, either using two Robosense Bpearl (*67*) dome LiDAR sensors or four Intel RealSense D435 depth cameras (*68*). We trained our policy in PyTorch (*69*) and deployed on the robot zero-shot without any fine-tuning. We build a robot-centric 2.5D elevation map at 20 Hz by estimating the robot's pose and registering the point-cloud readings from the sensors accordingly. The policy runs at 50 Hz and samples the heights from the latest elevation map, filling a randomly sampled value if no map information is available at a query location.

We developed an elevation mapping pipeline for fast terrain mapping on a graphics processing unit to parallelize point-cloud processing. We follow a similar approach to that used by Fankhauser *et al.* (*17*) to update the map in a Kalman filter fashion and additionally perform drift compensation and ray casting to obtain a more consistent map. This fast mapping implementation was crucial to maintain fast processing rates and keep up with the fast locomotion speeds achieved by our controller.

SUPPLEMENTARY MATERIALS

www.science.org/doi/10.1126/scirobotics.abk2822
Sections S1 to S9
Figs. S1 and S2
Tables S1 to S5
Movies S1 to S4

REFERENCES AND NOTES

1. M. Raibert, K. Blankespoor, G. Nelson, R. Playter, Bigdog, the rough-terrain quadruped robot. *IFAC Proceedings Volumes* **41**, 10822–10825 (2008).
2. B. Katz, J. Di Carlo, S. Kim, Mini cheetah: A platform for pushing the limits of dynamic quadruped control, in *2019 International Conference on Robotics and Automation (ICRA)* (IEEE, 2019), pp. 6295–6301.
3. 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**, 26 (2019).
4. J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, Learning quadrupedal locomotion over challenging terrain. *Sci. Robot.* **5**, abc5986 (2020).
5. H.-W. Park, P. M. Wensing, S. Kim, Jumping over obstacles with MIT Cheetah 2. *Robot. Auton. Syst.* **136**, 103703 (2021).
6. Boston Dynamics, Spot (2021); www.bostondynamics.com/spot [online; accessed March 2021].
7. C. Gehring, P. Fankhauser, L. Isler, R. Diethelm, S. Bachmann, M. Potz, L. Gerstenberg, M. Hutter, ANYmal in the field: Solving industrial inspection of an offshore HVDC platform with a quadrupedal robot, in *Field and Service Robotics* (Springer, 2021), pp. 247–260.
8. Agility Robotics, Robots (2021); www.agilityrobotics.com/robots [online; accessed June 2021].
9. Unitree Robotics, A1 (2021); www.unitree.com/products/a1/ [online; accessed March 2021].
10. Ghost Robotics, Vision 60 (2021); www.ghostrobotics.io/ [online; accessed June 2021].
11. J. S. Matthis, J. L. Yates, M. M. Hayhoe, Gaze and the control of foot placement when walking in natural terrain. *Curr. Biol.* **28**, 1224–1233 (2018).
12. ANYbotics, ANYmal (2021); www.anybotics.com/anymal-autonomous-legged-robot/ [online; accessed June 2021].
13. P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, R. Siegwart, Kinect v2 for mobile robot navigation: Evaluation and modeling, in *2015 International Conference on Advanced Robotics (ICAR)* (IEEE, 2015), pp. 388–394.
14. C. Ye, J. Borenstein, A new terrain mapping method for mobile robots obstacle negotiation, in *Unmanned ground vehicle technology V* (International Society for Optics and Photonics, 2003), pp. 52–62.
15. D. Belter, P. Skrzypczyński, Rough terrain mapping and classification for foothold selection in a walking robot, in *2010 IEEE Safety Security and Rescue Robotics*, Bremen, Germany, 26 to 30 July 2010 (IEEE, 2010), pp. 1–6.
16. P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, R. Siegwart, Robot-centric elevation mapping with uncertainty estimates, in *Mobile Service Robotics* (World Scientific, 2014), pp. 433–440.
17. P. Fankhauser, M. Bloesch, M. Hutter, Probabilistic terrain mapping for mobile robots with uncertain localization. *IEEE Robot. Autom. Lett.* **3**, 3019–3026 (2018).
18. M. Zucker, J. A. Bagnell, C. G. Atkeson, J. Kuffner, An optimization approach to rough terrain locomotion, in *2010 IEEE International Conference on Robotics and Automation* (IEEE, 2010), pp. 3589–3595.
19. P. D. Neuhaus, J. E. Pratt, M. J. Johnson, Comprehensive summary of the institute for human and machine cognition's experience with LittleDog. *Int. J. Robot. Res.* **30**, 216–235 (2011).
20. J. Z. Kolter, Y. Kim, A. Y. Ng, Stereo vision and terrain modeling for quadruped robots, in *2009 IEEE International Conference on Robotics and Automation* (IEEE, 2009), pp. 1557–1564.
21. I. Havoutis, J. Ortiz, S. Bazeille, V. Barasuol, C. Semini, D. G. Caldwell, Onboard perception-based trotting and crawling with the hydraulic quadruped robot (HyQ), in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2013), pp. 6052–6057.
22. C. Mastalli, M. Focchi, I. Havoutis, A. Radulescu, S. Calinon, J. Buchli, D. G. Caldwell, C. Semini, Trajectory and foothold optimization using low-dimensional models for rough terrain locomotion, in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2017), pp. 1096–1103.
23. D. Belter, P. Łabęcki, P. Skrzypczyński, Adaptive motion planning for autonomous rough terrain traversal with a walking robot. *J. Field Robot.* **33**, 337–370 (2016).
24. P. Fankhauser, M. Bjelonic, C. D. Bellicoso, T. Miki, M. Hutter, Robust rough-terrain locomotion with a quadrupedal robot, in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2018), pp. 5761–5768.
25. F. Jenelten, T. Miki, A. E. Vijayan, M. Bjelonic, M. Hutter, Perceptive locomotion in rough terrain–online foothold optimization. *IEEE Robot. Autom. Lett.* **5**, 5370–5376 (2020).
26. D. Kim, D. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, 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)* (IEEE, 2020), pp. 2464–2470.
27. O. A. Villarreal-Magaña, V. Barasuol, M. Camurri, M. Focchi, L. Franceschi, M. Pontil, D. G. Caldwell, C. Semini, Fast and continuous foothold adaptation for dynamic locomotion through CNNs. *IEEE Robot. Autom. Lett.* **4**, 2140–2147 (2019).
28. Boston Dynamics, Atlas | partners in parkour (2021); https://youtu.be/tF4DML7FIWk [online; accessed September 2021].
29. X. B. Peng, G. Berseth, M. Van de Panne, Terrain-adaptive locomotion skills using deep reinforcement learning. *ACM Trans. Graph.* **35**, 1–12 (2016).
30. X. B. Peng, G. Berseth, K. Yin, M. Van De Panne, Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Trans. Graph.* **36**, 1–13 (2017).
31. X. B. Peng, P. Abbeel, S. Levine, M. van de Panne, Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph.* **37**, 143:1–143:14 (2018).
32. Z. Xie, H. Y. Ling, N. H. Kim, M. van de Panne, Allsteps: Curriculum-driven learning of stepping stone skills, in *Computer Graphics Forum* (Wiley Online Library, 2020), pp. 213–224.
33. V. Tsounis, M. Alge, J. Lee, F. Farshidian, M. Hutter, Deepgait: Planning and control of quadrupedal gaits using deep reinforcement learning. *IEEE Robot. Autom. Lett.* **5**, 3699–3706 (2020).
34. J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, V. Vanhoucke, Sim-to-real: Learning agile locomotion for quadruped robots, in *Robotics: Science and Systems*, Pittsburgh, PA, USA, 26 to 30 June 2018 (2018).
35. X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, S. Levine, Learning agile robotic locomotion skills by imitating animals, in *Robotics: Science and Systems* (2020).
36. Y. Yang, K. Caluwaerts, A. Iscen, T. Zhang, J. Tan, V. Sindhwani, Data efficient reinforcement learning for legged robots, in *Conference on Robot Learning* (PMLR, 2020), pp. 1–10.
37. Z. Xie, P. Clary, J. Dao, P. Morais, J. Hurst, M. van de Panne, Learning locomotion skills for cassie: Iterative design and sim-to-real, in *Proceedings of the Conference on Robot Learning*, L. P. Kaelbling, D. Kragic, K. Sugiura, Eds. (PMLR, 2020), pp. 317–329.
38. J. Siekmann, K. Green, J. Warila, A. Fern, J. Hurst, Blind bipedal stair traversal via sim-to-real reinforcement learning, in *Robotics: Science and Systems* (2021).
39. A. Kumar, Z. Fu, D. Pathak, J. Malik, Rma: Rapid motor adaptation for legged robots, in *Robotics: Science and Systems* (2021).
40. C. Yang, K. Yuan, Q. Zhu, W. Yu, Z. Li, Multi-expert learning of adaptive legged locomotion. *Sci. Robot.* **5**, eabb2174 (2020).
41. J. Lee, J. Hwangbo, M. Hutter, Robust recovery controller for a quadrupedal robot using deep reinforcement learning. arXiv:1901.07517 (2019).

42. S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, I. Havoutis, RLOC: Terrain-aware legged locomotion using reinforcement learning and optimal control. arXiv:2012.03094 (2020).
43. M. Focchi, R. Orsolino, M. Camurri, V. Barasuol, C. Mastalli, D. G. Caldwell, C. Semini, Heuristic planning for rough terrain locomotion in presence of external disturbances and variable perception quality, in *Advances in Robotics Research: From Lab to Market* (Springer, 2020), pp. 165–209.
44. Boston Dynamics, Spot user guide release 2.0 version A (2021); www.generationrobots.com/media/spot-boston-dynamics/spot-user-guide-r2.0-va.pdf [online; accessed June 2021].
45. D. Chen, B. Zhou, V. Koltun, P. Krähenbühl, Learning by cheating, in *Conference on Robot Learning* (PMLR, 2020), pp. 66–75.
46. M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, R. Siegwart, State estimation for legged robots-consistent fusion of leg kinematics and IMU. *Robotics* **17**, 17–24 (2013).
47. Komoot, Etzel kulm loop hike (2021); https://bit.ly/35bjfyE [online; accessed June 2021].
48. Komoot, Komoot help guides (2021); https://d21buns5ku92am.cloudfront.net/67683/documents/40488-Komoot [online; accessed December 2021].
49. R. C. Coulter, Implementation of the pure pursuit path tracking algorithm, *Tech. rep.*, Carnegie-Mellon UNIV Pittsburgh PA Robotics INST (1992).
50. M. Tranzatto, F. Mascarich, L. Bernreiter, C. Godinho, M. Camurri, S. M. K. Khattak, T. Dang, V. Reijgwart, J. Loeje, D. Wisth, S. Zimmermann, H. Nguyen, M. Fehr, L. Solanka, R. Buchanan, M. Bjelonic, N. Khedekar, M. Valceschini, F. Jenelten, M. Dharmadhikari, T. Homberger, P. De Petris, L. Wellhausen, M. Kulkarni, T. Miki, S. Hirsch, M. Montenegro, C. Papachristos, F. Tresoldi, J. Carius, G. Valsecchi, J. Lee, K. Meyer, X. Wu, J. Nieto, A. Smith, M. Hutter, R. Y. Siegwart, M. Mueller, M. Fallon, K. Alexis, CERBERUS: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge, *Journal of Field Robotics* (2021).
51. Cerberus, Team cerberus (2021); www.subt-cerberus.org/ [online; accessed June 2021].
52. DARPA, Darpa subterranean challenge competition results finals (2021); www.subtchallenge.com/results.html [online; accessed November 2021].
53. DARPA, Darpa subterranean challenge competition rules final event (2021); www.subtchallenge.com [online; accessed June 2021].
54. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, *Advances in Neural Information Processing Systems*, *Deep Learning Workshop* (2013).
55. P. Zhu, X. Li, P. Poupart, G. Miao, On improving deep reinforcement learning for pomdps. arXiv:1704.07978 (2017).
56. O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. M. Kinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, D. Silver, Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature* **575**, 350–354 (2019).
57. S. Bai, J. Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271 (2018).
58. J. Hwangbo, J. Lee, M. Hutter, Per-contact iteration method for solving contact dynamics. *IEEE Robot. Autom. Lett.* **3**, 895–902 (2018).
59. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms. arXiv:1707.06347 (2017).
60. S. Ross, G. Gordon, J. 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), pp. 627–635.
61. W. M. Czarnecki, R. Pascanu, S. Osindero, S. Jayakumar, G. Swirszcz, M. Jaderberg, Distilling policy distillation, in *Proceedings of Machine Learning Research*, K. Chaudhuri, M. Sugiyama, Eds. (PMLR, 2019), pp. 1331–1340.
62. K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, in *Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014), pp. 1724–1734.
63. S. Hochreiter, J. Schmidhuber, Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
64. T. Anzai, K. Takahashi, Deep gated multi-modal learning: In-hand object pose changes estimation using tactile and image data, in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2020), pp. 9361–9368.
65. J. Kim, J. Koh, Y. Kim, J. Choi, Y. Hwang, J. W. Choi, Robust deep multi-modal learning based on gated information fusion network, in *Asian Conference on Computer Vision* (Springer, 2019), pp. 90–106.
66. J. Arevalo, T. Solorio, M. Montes-y Gómez, F. A. González, Gated multimodal units for information fusion, *ICLR workshop* (2017).
67. Rs-bpearl (April 2021); www.robosense.ai/en/rslidar/RS-Bpearl.
68. Intel RealSense (April 2021); www.intelrealsense.com/.
69. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, R. Garnett, Eds. (Curran Associates Inc., 2019), pp. 8024–8035.

Funding: The project was funded, in part, by the Intel Network on Intelligent Systems, the Swiss National Science Foundation (SNF) through the National Centre of Competence in Research Robotics and project no. 188596, and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme grant agreement nos. 852044, 780883, and 101016970. This work has been conducted as part of ANYmal Research, a community to advance legged robotics. **Author contributions:** T.M. formulated the main idea of combining inputs from multiple modalities. J.L. and J.H. designed and tested the initial setup. T.M. developed software and trained the controller. T.M. and L.W. set up the perception pipeline on the robot. T.M. conducted most of the indoor experiments. T.M., J.L., and L.W. conducted outdoor experiments. All authors refined ideas, contributed in the experiment design, analyzed the data, and wrote the paper. **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 5 July 2021
Accepted 20 December 2021
Published 19 January 2022
10.1126/scirobotics.abk2822

Science Robotics

Learning robust perceptive locomotion for quadrupedal robots in the wild

Takahiro MikiJoonho LeeJemin HwangboLorenz WellhausenVladlen KoltunMarco Hutter

Sci. Robot., 7 (62), eabk2822. • DOI: 10.1126/scirobotics.abk2822

Science Robotics (ISSN) 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.