Science Robotics
AAAS

Supplementary Materials for

Learning robust perceptive locomotion for quadrupedal robots in the wild

Takahiro Miki *et al.*

Corresponding author: Takahiro Miki, tamiki@ethz.ch

Sci. Robot. **7**, eabk2822 (2022)
DOI: 10.1126/scirobotics.abk2822

The PDF file includes:

Sections S1 to S9
Figs. S1 and S2
Tables S1 to S5

Other Supplementary Material for this manuscript includes the following:

Movies S1 to S4

S1. Nomenclature

s	state
o	observation
b	belief state
h	hidden state
l	latent feature
v	linear velocity
ω	angular velocity
τ	joint torque
q	joint position
ϕ	CPG phase
$\Delta\phi_0$	CPG phase base frequency
c_k	curriculum factor
c_{sk}	student curriculum factor
$\mathcal{L}_{bc}$	behavior cloning loss
$\mathcal{L}_{re}$	reconstruction loss
$(\cdot)^p$	proprioceptive quantity
$(\cdot)^e$	exteroceptive quantity
$(\cdot)^{priv}$	privileged quantity
$(\cdot)^{target}$	target quantity
$(\cdot)_t$	quantity at time t
$\tilde{(\cdot)}$	noisy quantity
$\dot{(\cdot)}$	first derivative
$\ddot{(\cdot)}$	second derivative
$g(\cdot)$	Multilayer Perceptron (MLP) encoder

$\mathcal{N}()$	Normal distribution
$\odot$	Hadamard product
$\mathbf{p}(\cdot)$	foot trajectory function
$IK(\cdot)$	inverse kinematics function

S2. Evaluating the importance of exteroception: Additional experiments in simulation

We compare the success rate over various stepped terrain and stairs in simulation to further evaluate the performance quantitatively.

The robot was given a fixed forward velocity command of 0.7 m/s for a duration of 10 seconds. We collected 300 trials to calculate the success rate, where we consider a trial a success if the robot can traverse 4 m without failure. As shown in Figure 8A, 8B our controller significantly outperformed the baseline and can traverse a much wider range of terrain.

S3. Training details

The control frequency of the policy was set to 50 Hz, and 250 trajectory time steps per environment are collected for one training iteration. We parallelized the simulation environment to perform rollouts with 1000 environments simultaneously. We used our custom implementation of PPO (*58*) to train the teacher policy (*69*). Observations are normalized using running mean and standard deviation before giving them to the policy network. The curriculum factors were updated exponentially every training episode $c_{k+1} = c_k^d$, with convergence rate $d = 0.98$. We use the Adam (*70*) optimizer with exponential learning rate decay. The hyperparameters for PPO are given in Table S1.

For student training, we performed rollouts with 300 environments and collected 400 timesteps of trajectory for one training iteration. We start the student training without height sample noise

Table 1: Hyperparameters for PPO.

learning rate	5.0 E-4
learning rate decay gamma	0.9999
discount factor	0.996
learning epoch	2
GAE-lambda	0.95
clip ratio	0.2
entropy coefficient	0.005
batch size	8300

Table 2: Hyperparameters for student training.

learning rate	5.0 E-4
truncate step for TBPTT	10
learning epoch	2

and gradually increase the noise level through a student curriculum factor which linearly increases over training epochs. We use flat terrain for the first 10 epochs, and then enable the adaptive curriculum for the terrain generation. After 20 epochs, we increase the student curriculum factor c_{sk} linearly until we reach 100 epochs. Then, we keep $c_{sk} = 1.0$. We train the RNN unit of the encoder with Truncated Backpropagation Through Time (TBPTT). The ratio between behavior cloning loss and reconstruction loss is 0.5. Therefore the loss is set to $\mathcal{L}_{bc} + 0.5 \cdot \mathcal{L}_{re}$. Hyperparameters for student training are given in Table S2.

S4. Terrain generation

The terrain types are *rough*, *rough discrete*, *large steps*, *boxes*, *grid steps*, *step stairs*, and *stairs*, as shown in Figure 6.1. There are four types of stairs: *standard stair*, *open stair*, *ledged stair*, and *random stair*. Each terrain type is parameterized by different terrain properties, which are randomized during training.

The *rough* terrain is parameterized by Perlin noise (*71*) and the *rough discrete* and *large steps* are created by quantizing it. While *rough discrete* terrain does not restrict the number of

quantization levels, *large steps* only allow for two height levels ($h \in [0, 0.4]$ m). For *grid steps*, the parameters are mean step height ($h \in [0.05, 0.4]$ m) and step width ($d \in [0.2, 0.7]$ m). Some examples of different *grid steps* are shown in Figure 8A. Note that the parameter range shown in the figure is only for evaluation and different from the range used during training. Parameters for *stairs* contain step depth ($d \in [0.25, 1.0]$ m) and height ($h \in [0.01, 0.22]$ m). The height and depth values for *random stair* were set at each according to a ratio $\epsilon \sim \mathcal{N}(1.0, 0.2)$, such that $\hat{x} = x \cdot \epsilon$, where x is the given depth or height parameter. Examples of different stairs are shown in Figure 8B. The *boxes* terrain consists of multiple boxes with maximum height 0.25 m lying in a random position with random yaw angles.

S5. Observation and action

The observation vectors are defined in Table S3. Proprioceptive input includes command, joint, and body information, as well as leg phase information. The central pattern generator (CPG)'s phase information consists of $\Delta\phi_l$, $\cos\phi_l$, $\sin\phi_l$, and base frequency for each leg l. For exteroception, we use height samples around each foot instead of the local elevation map. The circular sampling pattern comprises {6, 8, 10, 12, 16} points around each foot, with radii {0.08, 0.16, 0.26, 0.36, 0.48} m, respectively.

The action is defined as $\langle \Delta\phi_l, \Delta q_i \rangle$, where $\Delta\phi_l$ and Δq_i refer to the phase offset per leg ($l \in \{legs\}$) and the residual joint position target ($i \in \{1, \cdots, 12\}$), respectively. We have a nominal foot trajectory $\mathbf{p}(\phi) : \mathbb{R} \longrightarrow \mathbb{R}^3$ that maps each ϕ_l to a target foot position, which generates periodic stepping motion as ϕ cycles within $[0, 2\pi)$. From the action, the joint position target for a leg l is defined as $q_{i\in l}^{target} = IK(\mathbf{p}(\phi_l + \Delta\phi_l + \Delta\phi_0)) + \Delta q_{i\in l}$, using analytic inverse kinematics $IK(\cdot)$ and base phase frequency $\Delta\phi_0$. The nominal foot trajectory is defined as follows.

Table 3: Observations. Proprioception is used for both teacher and student training. Exteroception is given in the form of height samples. The privileged information is used only for teacher training.

Observation type	Input	Dimensionality
Proprioception	command	3
	body orientation	3
	body velocity	6
	joint position	12
	joint velocity	12
	joint position history (3 time steps)	36
	joint velocity history (2 time steps)	24
	joint target history (2 time steps)	24
	CPG phase information	13
Exteroception	height samples	208
Privileged info.	contact states	4
	contact forces	12
	contact normals	12
	friction coefficients	4
	thigh and shank contact	8
	external forces and torques	6
	airtime	4

If the phase is in swing-up ($0 \leq \phi_l \leq \pi/2$),

$$\begin{aligned} \mathbf{p}_l(\phi_l) \quad &= \langle x_l^n, y_l^n, z_l^n + 0.2 \cdot (-2t_l^3 + 3t_l^2) \rangle, \\ &\text{where} \quad t_l = 2/\pi \cdot \phi_l. \end{aligned}$$

$\{x, y, z\}_l^n$ is the nominal foot position at the default stance configuration. The cubic Hermite spline connects $z = z_l^n$ at $\phi_l = 0$ and $z = z_l^n + 0.2$ at $\phi_l = \pi/2$.

In the swing-down phase ($\pi/2 < \phi_l \leq \pi$), the foot height is computed as

$$\begin{aligned} \mathbf{p}_l(\phi_l) \quad &= \langle x_l^n, y_l^n, z_l^n + 0.2 \cdot (2t_l^3 - 3t_l^2 + 1) \rangle, \\ &\text{where} \quad t_l = 2/\pi \cdot \phi_l - 1, \end{aligned}$$

which is symmetric to the previous function.

During the stance phase ($\pi < \phi_l \leq 2\pi$), $\mathbf{p}_l(\phi_l) = \langle x_l^n, y_l^n, z_l^n \rangle$.

S6. Network architecture

The policy network is composed of multiple MLPs. The height samples are first encoded into a $24 \times 4 = 96$ dimensional latent vector, and the privileged information is encoded into a 24 dimensional latent vector using MLP-based encoders (g_e, g_p). Each encoder has two hidden layers with $\{80, 60\}$ and $\{64, 32\}$ hidden units respectively. The height samples are first fed into the encoder separately for each foot and then concatenated into one feature vector. Then these features are concatenated with proprioceptive observations and fed into another MLP with three hidden layers $\{256, 160, 128\}$. The activation function for all MLPs is LeakyReLU (*72*).

We use a GRU with an exteroceptive gate for the belief encoder (Figure 7C). The GRU consists of 2 stacked layers with 50 hidden units each. The belief encoder and exteroceptive gate g_b, g_a are used to calculate $96 + 24 = 120$ dimensional belief state b_t and 96 dimensional attention vector α. Each encoder has two hidden layers with $\{64, 64\}$ and $\{64, 64\}$ hidden

units each. The filtered exteroceptive information $l_t^e \odot \alpha$ is added to $g_b(b'_t)$, with zero-padding to match the dimensionality.

S7. Reward function

The reward function is defined as $r = 0.75(r_{lv} + r_{av} + r_{lvo}) + r_b + 0.003r_{fc} + 0.1r_{co} + 0.001r_j + 0.08r_{jc} + 0.003r_s + 1.0 \cdot 10^{-6}r_\tau + 0.003r_{slip}$. The individual terms are defined as follows.

- Linear Velocity Reward (r_{lv}): This term encourages the policy to follow a desired horizontal velocity (velocity in xy plane) command:

$$r_{lv} = \begin{cases} \exp(-|\boldsymbol{v}|^2), & \text{if } |\boldsymbol{v}_{des}| = 0 \\ 1.0, & \text{else if } \boldsymbol{v}_{des} \cdot \boldsymbol{v} > |\boldsymbol{v}_{des}| \\ \exp(-(\boldsymbol{v}_{des} \cdot \boldsymbol{v} - |\boldsymbol{v}_{des}|)^2), & \text{otherwise} \end{cases}$$

  where $\boldsymbol{v}_{des} \in \mathbb{R}^2$ is the desired horizontal velocity and $\boldsymbol{v} \in \mathbb{R}^2$ is the current body velocity with respect to the body frame.

- Angular Velocity Reward (r_{av}): This term encourages the policy to follow a desired yaw velocity command:

$$r_{av} = \begin{cases} \exp(-\omega_z^2), & \text{if } \omega_{des} = 0 \\ 1.0, & \text{else if } \omega_{des} \cdot \omega_z > \omega_{des} \\ \exp(-(\omega_{des} \cdot \omega_z - \omega_{des})^2), & \text{otherwise} \end{cases}$$

  where ω_{des} is the desired yaw velocity and ω_z is the current yaw velocity with respect to the body frame.

- Linear Orthogonal Velocity Reward (r_{lvo}): This term penalizes the velocity orthogonal to the target direction:

$$r_{lvo} = \exp(-3.0|\boldsymbol{v}_o|^2),$$

  where $\boldsymbol{v_o} = \boldsymbol{v} - (\boldsymbol{v}_{des} \cdot \boldsymbol{v})\boldsymbol{v}_{des}$.

- Body motion Reward (r_b): This term penalizes the body velocity in directions not part of the command:

$$r_{bm} = -1.25v_z^2 - 0.4|\omega_x| - 0.4|\omega_y|.$$

- Foot Clearance Reward (r_{fc}): When a leg is in swing phase, i.e., $\phi_i \in [0, \pi)$, the robot should lift the corresponding foot higher than its surroundings. However, to prevent the robot from manifesting unnecessarily high foot clearance, we give a penalty reward r_{fcl} to regularize the leg trajectory. $H_{sample,l}$ is the set of sampled heights around the l-th foot. Then, the clearance cost is defined as

$$\begin{aligned} r_{fcl} &= \begin{cases} -1.0, & \text{if } \max(H_{sample,l}) < -0.2 \\ 0.0 & \text{otherwise} \end{cases} \\ r_{fc} &= \sum_{l=1}^{4} r_{fcl} \end{aligned}$$

  Note that height samples are sampled with respect to the foot height, therefore -0.2 means the terrain is 0.2 m lower than the foot; ergo, the foot is 0.2 m higher than the sampled terrain height.

- Shank and Knee Collision Reward (r_{co}): We want to penalize undesirable contact between the terrain and robot parts other than the foot, to avoid hardware damage:

$$r_{co} = \begin{cases} -c_k, & \text{if shank or knee is in collision} \\ 0.0 & \text{otherwise} \end{cases}$$

  where c_k is the curriculum factor that increases monotonically and converges to 1.

- Joint Motion Reward (r_j): This term penalizes joint velocity and acceleration to avoid vibrations:

$$r_s = -c_k \sum_{i=1}^{12} (0.01\dot{q_i}^2 + \ddot{q_i}^2),$$

  where $\dot{q_i}$ and $\ddot{q_i}$ are the joint velocity and acceleration, respectively.

- Joint Constraint Reward (r_{jc}): This term introduces a soft constraint in the joint space. To avoid the knee joint flipping in the opposite direction, we give a penalty for exceeding a threshold:

$$\begin{aligned} r_{jc,i} &= \begin{cases} -(q_i - q_{i,th})^2, & \text{if } q_i > q_{i,th} \\ 0.0 & \text{otherwise} \end{cases} \\ r_{jc} &= \sum_{i=1}^{12} r_{jc,i} \end{aligned}$$

where $q_{i,th}$ is a threshold value for the ith joint. We only set thresholds for the knee joint.

- Target Smoothness Reward (r_s): The magnitude of the first and second order finite difference derivatives of the target foot positions are penalized such that the generated foot trajectories become smoother:

$$r_s = -c_k \sum_{i=1}^{12} ((q_{i,t}^{des} - q_{i,t-1}^{des})^2 + (q_{i,t}^{des} - 2q_{i,t-1}^{des} + q_{i,t-2}^{des})^2),$$

where $q_{i,t}^{des}$ is the joint target position of joint i at time step t.

- Torque Reward (r_τ): We penalize joint torques to reduce energy consumption ($\tau \propto$ electric current):

$$r_\tau = -c_k \sum_{i=1}^{12} \tau_i^2,$$

where τ_i is the ith joint's torque calculated as output by the actuator network.

- Slip Reward (r_{slip}): We penalize the foot velocity if the foot is in contact with the ground to reduce slippage:

$$r_{slip} = -c_k \sum_{l \in \{\text{foot in contact}\}} v_{f,l}^2,$$

where $v_{f,l}$ is the velocity of lth foot in contact with the ground.

S8. Height sample noise

During student training, we randomize the height samples drawn around each foot (Figure 7A). We perturbed the position of each sample and add noise to the measured height value as follows.

$$x_p = r_p \cos(\theta_p) + \epsilon_{px} + \epsilon_{fx} + w_x$$

$$y_p = r_p \sin(\theta_p) + \epsilon_{py} + \epsilon_{fy} + w_y$$

$$h_p = h(x_p, y_p) + \epsilon_{pz} + \epsilon_{fz} + w_z + \epsilon_{outlier}$$

where $h(x_p, y_p)$ refers to the terrain height at position (x_p, y_p). r_p is the radial distance of the point p and θ_p is the azimuthal angle of p in polar coordinates around the foot. $\epsilon_{px}, \epsilon_{py}, \epsilon_{pz}$ represents the noise that is sampled for each individual point every time step. $\epsilon_{fx}, \epsilon_{fy}, \epsilon_{fz}$ represents the noise that is sampled for each foot every time step. w_x, w_y, w_z represents the noise that is sampled for each foot per episode. $\epsilon_{outlier}$ is a large noise intermittently added to simulate outliers.

Each noise is sampled from the normal distribution using the parameter z. $\epsilon_{px}, \epsilon_{py} \sim \mathcal{N}(0, z_0)$, $\epsilon_{pz} \sim \mathcal{N}(0, z_1)$, $\epsilon_{fx}, \epsilon_{fy} \sim \mathcal{N}(0, z_2)$, $\epsilon_{fz} \sim \mathcal{N}(0, z_3)$, $\epsilon_{outlier} \sim \mathcal{N}(0, z_4)$ with probability $p = z_5$, $w_x, w_y \sim \mathcal{N}(0, z_6)$, $w_z \sim \mathcal{N}(0, z_7)$.

We defined three conditions for the student training; *nominal*, *offset*, *noisy*. Each parameter z is defined as follows.

$$z_{nominal} = \langle 0.004, 0.005, 0.01, 0.04, 0.03, 0.05, 0.1 \rangle \quad (2)$$

$$z_{offset} = \langle 0.004, 0.005, 0.01, 0.1c_{sk}, 0.1c_{sk}, 0.02, 0.1 \rangle \quad (3)$$

$$z_{noisy} = \langle 0.004, 0.1c_{sk}, 0.1c_{sk}, 0.3c_{sk}, 0.3c_{sk}, 0.3c_{sk}, 0.1 \rangle \quad (4)$$

where c_{sk} is the student curriculum factor which linearly increases over training episodes. We randomly picked one of the conditions at the beginning and in the middle of a trajectory. The probabilities are 60%, 30% and 10%, respectively.

S9. Ablation study of attention gate in belief encoder

We evaluated the effect of the exteroceptive gate by comparing the performance of the belief encoder with and without the gate. For this purpose, we trained four student policies using different belief encoders: "GRU gate", "GRU no gate", "MLP gate" and "MLP no gate". "GRU gate" uses the proposed exteroceptive gate while "GRU no gate" does not use it. "MLP" uses feed forward network instead of the recurrent unit. Figure S2A shows the learning curve of the student training using four different architectures. The result shows that using a recurrent unit improves the performance. MLP failed to reconstruct the privileged information. Moreover, the exteroceptive gate constantly improves the performance for both GRU and MLP architectures. Note that in the beginning of the training, we started without exteroceptive noise and terrain curriculum, and increased them gradually. This effect can be seen as a steep increase of losses and decrease of reward in the beginning.

To evaluate the learned model, we collected 300 time steps with 100 different terrain parameters for each terrain type with two noise conditions: *small* and *large*. Each noise parameter z are defined as follows,

$$z_{small} = \langle 0.004, 0.005, 0.04, 0.04, 0.04, 0.01, 0.1 \rangle \quad (5)$$

$$z_{large} = \langle 0.004, 0.3, 0.2, 0.1, 0.1, 0.03, 0.1 \rangle \quad (6)$$

Then we calculated the squared distance between student action and teacher action, as well as decoded height samples and ground-truth height samples. As shown in Table S4, S5, the gated encoder outperformed the non-gated encoder for both noise cases. The encoder utilizes the exteroceptive input through the skip connection when the exteroception is reliable. When the height samples contain large noise, the exteroception does not provide reliable information. In this case, the gated structure and non-gated structure perform similarly (Table S4, S5). This indicates that the gated structure facilitates the use of exteroceptive information when it is reliable

Table 4: Action difference between teacher and student under different noise conditions. The quantities are presented as empirical means with standard deviations. The belief encoder with the exteroceptive gate exhibits smaller action difference for all types of terrain when the noise is small. When the exteroception is unreliable (large noise), they perform similarly; this indicates that the gate blocks the skip connection such that our encoder becomes similar to the proprioceptive model in this condition.

	Small exteroceptive noise		Large exteroceptive noise	
terrain	ours	without gate	ours	without gate
rough	**0.690±0.40**	0.746±0.40	**0.879±0.46**	0.997±0.44
rough discrete	**0.787±0.45**	0.857±0.54	**0.878±0.53**	0.964±0.55
step stair	**0.652±0.39**	0.687±0.43	**0.975±0.49**	1.043±0.50
large step	**0.719±0.40**	0.855±0.43	**1.142±0.55**	1.225±0.54
grid steps	**1.444±0.56**	1.674±0.58	2.218±0.70	**2.212±0.70**
standard stair	**0.854±0.67**	0.961±0.72	**1.387±0.59**	1.438±0.56
open stair	**0.842±0.61**	0.938±0.65	**1.356±0.55**	1.428±0.53
ledged stair	**0.819±0.39**	0.929±0.42	**1.373±0.53**	1.416±0.54
boxes	**0.928±0.53**	1.123±0.56	**1.614±0.64**	1.683±0.68
random stair	**0.872±0.45**	0.956±0.46	**1.489±0.59**	1.526±0.58

but does not sacrifice robustness when it becomes unreliable.

To further evaluate the policies' performance, a step traversal success rate were compared against each policy. The robot was initialized in front of various height of step and given a constant velocity command (0.8 m/s) towards the step. We collected 100 trials for each height of the step and showed the success rate in Figure S2B. The result shows that "GRU gate" performs the best for both small noise and large noise case. As seen in the small noise case, the difference between "GRU gate" and "GRU no gate" is bigger than the large noise case. This supports that the gated structure can utilize exteroceptive information more when it is reliable.

Table 5: Reconstruction error of height samples under different noise conditions. The quantities are presented as empirical means with standard deviations. The belief encoder with the exteroceptive gate had smaller reconstruction error for all types of terrain. This shows the effectiveness of the gated skip connection when the exteroception is reliable. When the noise is large, the gated encoder also performed better than the non-gated encoder, although the difference was smaller than in the small-noise setting.

	Small exteroceptive noise		Large exteroceptive noise	
terrain	ours	without gate	ours	without gate
rough	**1.21E-03±2.8E-04**	1.36E-03±6.1E-04	**1.03E-03±2.3E-04**	1.17E-03±5.9E-04
rough discrete	**9.99E-04±3.3E-04**	1.03E-03±3.9E-04	**1.02E-03±3.5E-04**	1.05E-03±3.5E-04
step stair	**1.13E-03±4.4E-04**	1.31E-03±4.7E-04	**1.41E-03±4.3E-04**	1.48E-03±4.6E-04
large step	**1.37E-03±8.0E-04**	2.03E-03±1.0E-03	**1.95E-03±8.2E-04**	**1.95E-03±7.8E-04**
grid steps	**3.05E-03±4.1E-04**	4.77E-03±7.4E-04	**4.17E-03±5.0E-04**	4.39E-03±5.1E-04
standard stair	**2.59E-03±2.2E-03**	3.11E-03±2.2E-03	**2.68E-03±1.6E-03**	2.69E-03±1.5E-03
open stair	**2.61E-03±2.3E-03**	3.06E-03±2.0E-03	**2.63E-03±1.2E-03**	2.64E-03±1.1E-03
ledged stair	**2.53E-03±1.7E-03**	3.03E-03±1.5E-03	**2.62E-03±1.2E-03**	2.63E-03±1.1E-03
boxes	**2.13E-03±1.4E-03**	3.38E-03±1.5E-03	**3.00E-03±1.0E-03**	3.09E-03±1.2E-03
random stair	**2.31E-03±9.1E-04**	2.89E-03±8.2E-04	**2.72E-03±7.9E-04**	2.74E-03±8.0E-04

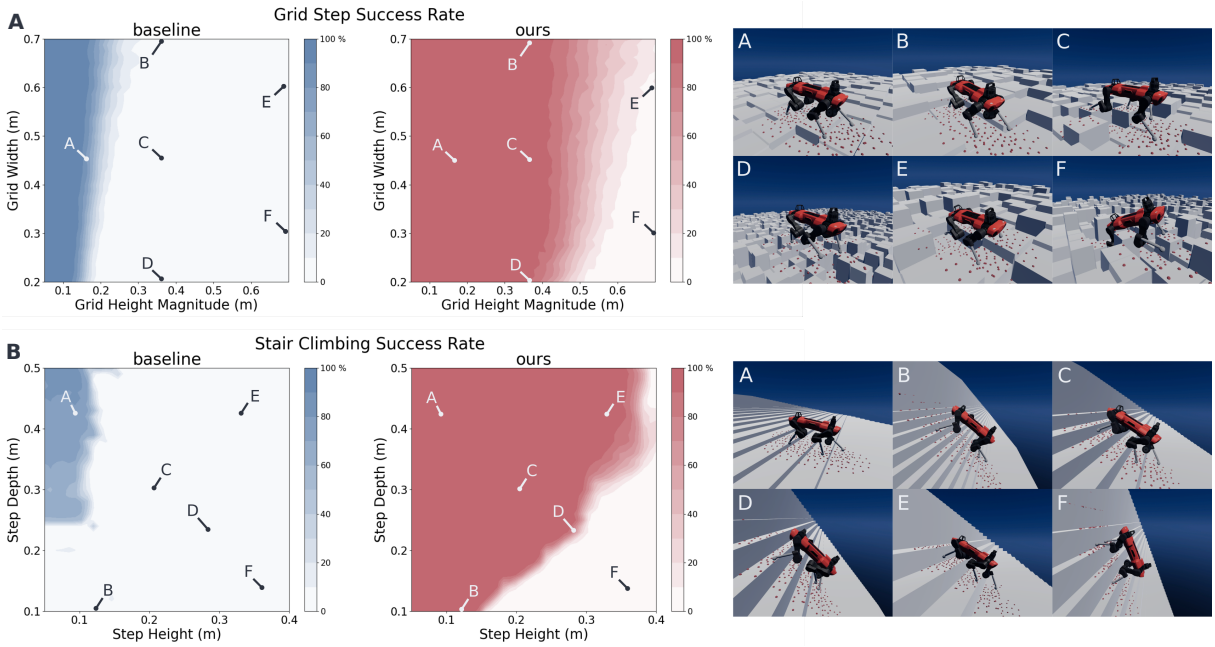

Figure S1: Comparison of the presented controller to a proprioceptive baseline (*4*) over random terrain. We collected 300 trials with a fixed velocity command over 41×41 different terrain parameter combinations and compared success rates. Our controller was able to traverse a much wider range of terrain profiles on both grid steps (A) and stairs (B).

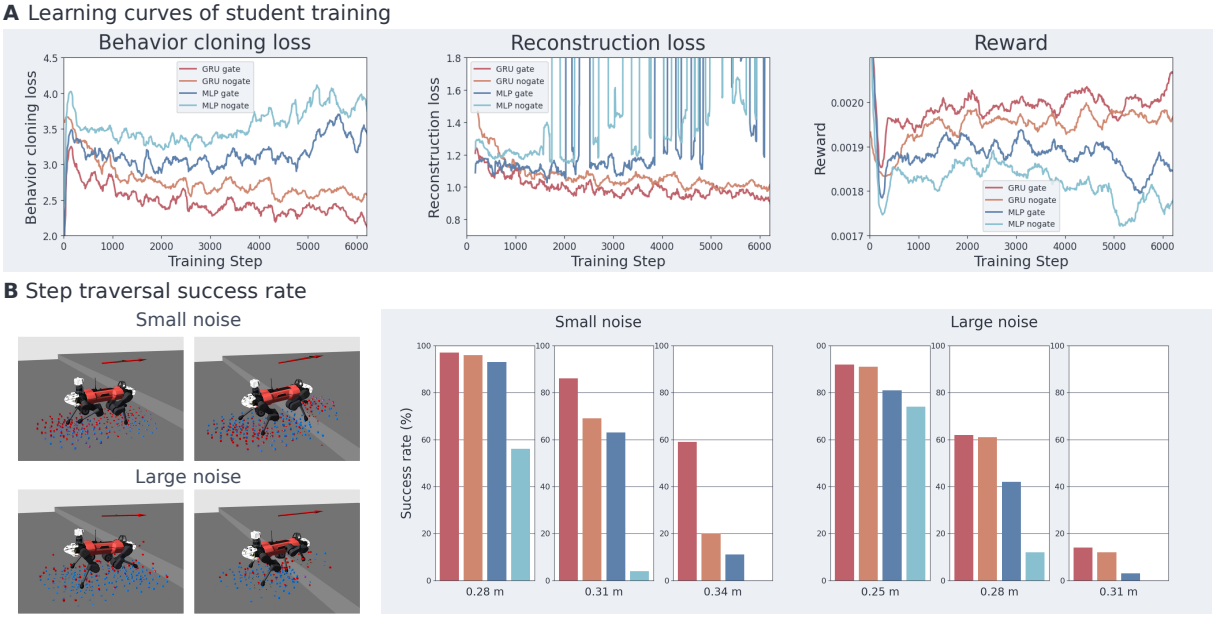

Figure S2: Ablation analysis of the presented belief encoder. We compared GRU gate, GRU no gate, MLP gate and MLP no gate. MLP setting uses MLP instead of GRU as its encoder. Gate setting uses proposed attention gate while no gate setting exclude it.(A) Learning curve of the student policy training. GRU worked better than MLP in all cases. Attention gate worked better than without attention for both GRU and MLP. The increase of the losses and decrease of reward in the beginning is due to the curriculum. (B) Step traversal success rate tested in small noise and large noise cases. The robot is initialized with random joint configuration and initial velocity and given a constant command towards the step. If the robot traversed the step with both front and hind legs it is considered as success. 100 trials were conducted.