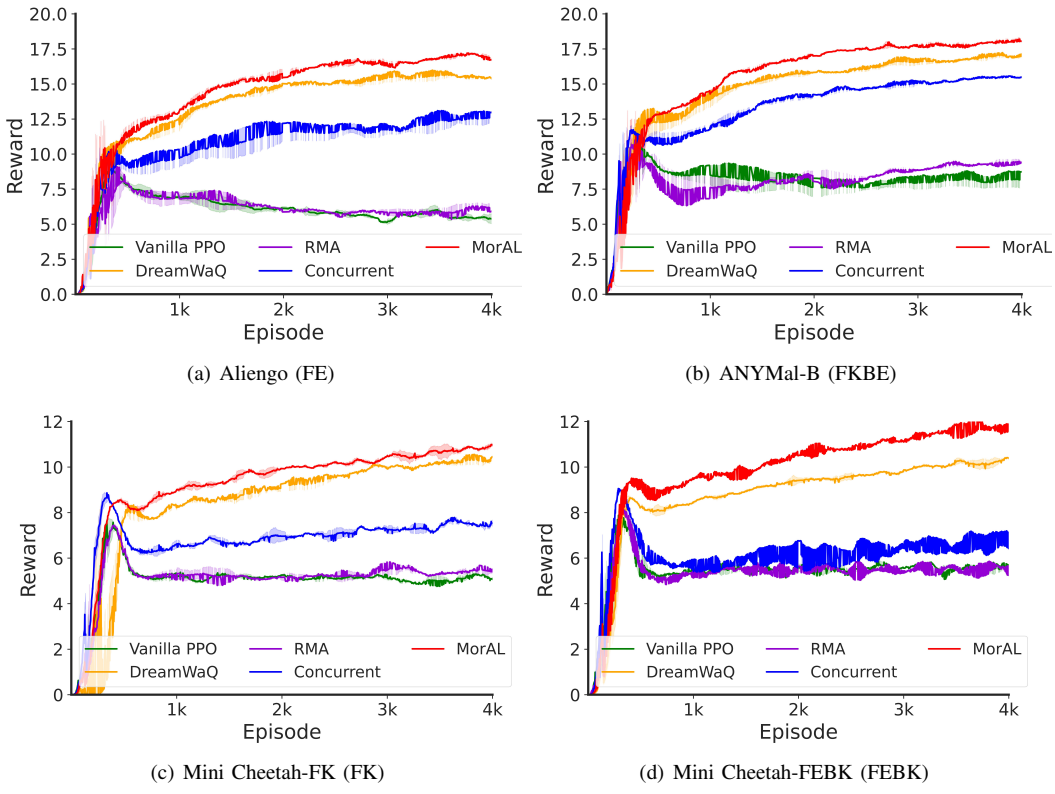
Supplementary Materials – MorAL: Learning Morphologically Adaptive Locomotion Controller for Quadrupedal Robots on Challenging Terrains

Abstract

This document is structured as follows. Section I demonstrates the comparative analysis between MorAL and other baseline algorithms in terms of their performance on robot-specific tasks. Section II presents the investigation of the impact of different hyperparameter β values on the training performance. This section also reports the converged regression loss of each component of the morph net's output. In Section III, we conduct a comprehensive evaluation of the actuator network's crucial role in addressing the sim-to-real transfer in this pipeline. The control policies trained with and without actuator net are compared. In Section IV, we provide the full details regarding the experiments of varying morphology, and a novel test demonstrating the adaptability to dynamic changes. Finally, several detailed explanations are complemented for the illustration in the main article.

I. Compared different Controllers on specific configurations

As the baseline algorithms listed in Section III-B are robot-specific, in order to make the comparison fairer, we compare the learning performance of these five different controllers on four specific robot configurations, including Aliengo (FE), ANYMal-B (FKBE), Mini Cheetah-FK (FK), and Mini Cheetah-FEBK (FEBK). Note that the MIT cheetah is one of the very few types of quadrupedal robot that possesses invertible thigh and calf joints and can form FK and FEBK due to its mechanical design. As shown in Fig. S1, DreamWaQ and Concurrent have better average rewards compared to Vanilla PPO and RMA. MorAL still exhibits the best performance in these four specific robot configurations over challenging terrains. This is attributed to MorAL not only acquiring the body velocity but also morphological information about the robot, which facilitates the control policy to reason about the robot's states.



(a) Aliengo (FE)

(b) ANYMal-B (FKBE)

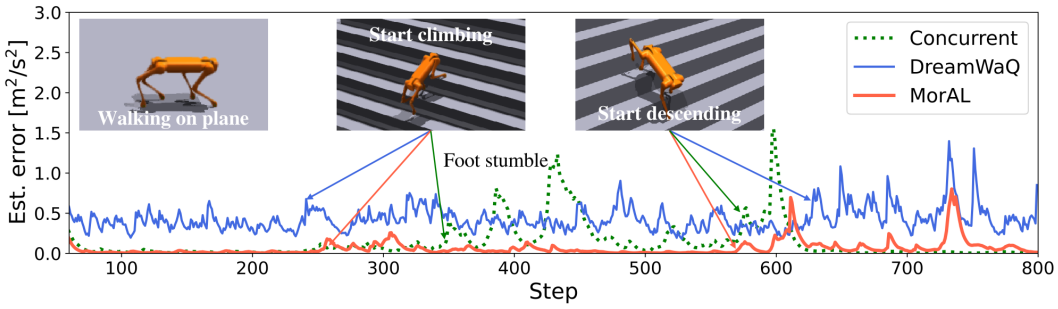
(c) Mini Cheetah-FK (FK)

(d) Mini Cheetah-FEBK (FEBK)

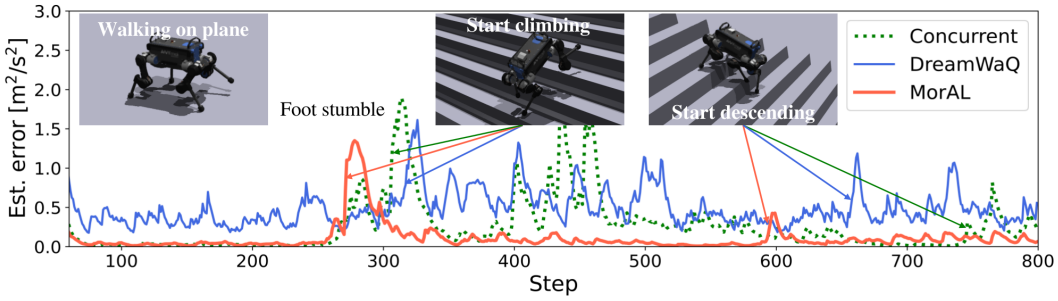
Fig. S1: The learning performance of different controllers over four specific configurations.

In addition, we also compared the velocity estimation errors of three different controllers (MorAL, Concurrent, and DreamWaQ) across four specific configurations. As shown in Fig. S2, these four robots are instructed to traverse a long path consisting of flat ground, ascending stairs, and descending stairs. For all configurations, the estimation error fluctuates over time and

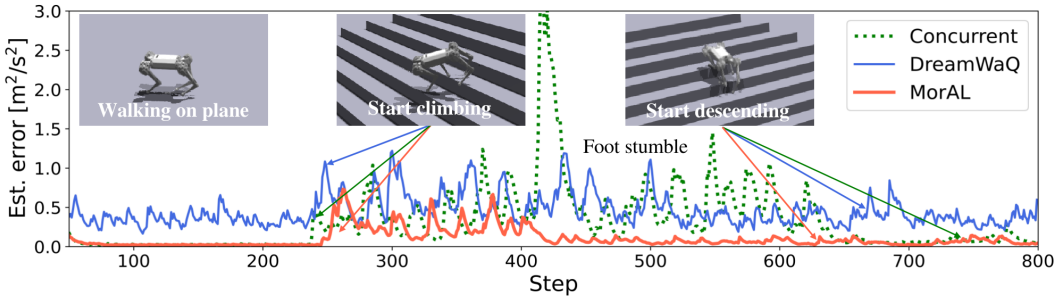
MorAL generally maintains a lower error compared to Concurrent and DreamWaQ. The impact of "foot stumble" on the velocity estimation error is particularly noticeable, with sharp spikes observed in the graphs. Although such perturbations are challenging for robots, MorAL still has the lowest error peak across all specific configurations and challenging tasks. Thus, it could be a more reliable and robust controller for velocity estimation in these scenarios.



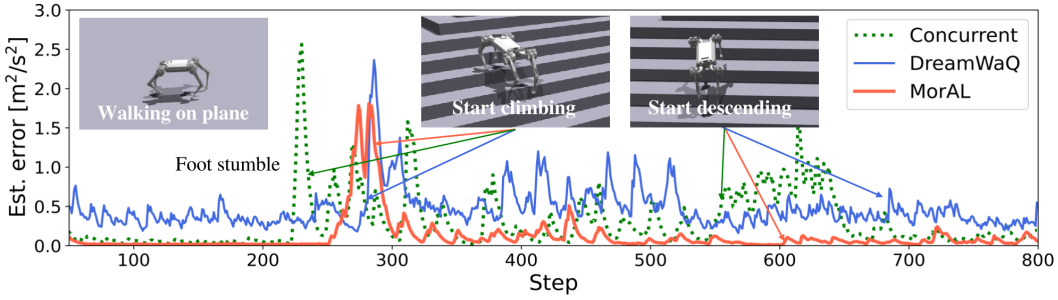
(a) Aliengo (FE)



(b) ANYMal-B (FKBE)



(c) Mini Cheetah-FK (FK)



(d) Mini Cheetah-FEBK (FEBK)

Fig. S2: Squared velocity estimation error of MorAL, Concurrent, and DreamWaQ over four specific configurations.

II. Evaluation on the Loss Function

Fig. S3 shows the impact of different weight $\beta \in [0, 1]$ on the training performance. The lines for $\beta = 0.1$ (orange), $\beta = 0.3$ (blue), $\beta = 0.5$ (purple), $\beta = 0.7$ (brown), and $\beta = 0.9$ (dark grey) exhibit a proximity, indicating comparable performance under these parameter configurations. Notably, they achieve higher rewards at a faster rate compared to $\beta = 0$ (green), which means that the morph net is only updated by the gradient of $loss_{policy}$ and not by the gradient of $loss_{reg}$. The line for $\beta = 1.0$

shows a very different pattern, with the reward remaining nearly constant and small throughout the episodes. This suggests that the setting of $\beta = 1.0$ is not conducive to the learning process. It is worth mentioning that the situation $\beta = 1.0$ means that the morph net is solely guided by regression loss $loss_{reg}$ and is independent of the policy loss $loss_{policy}$.

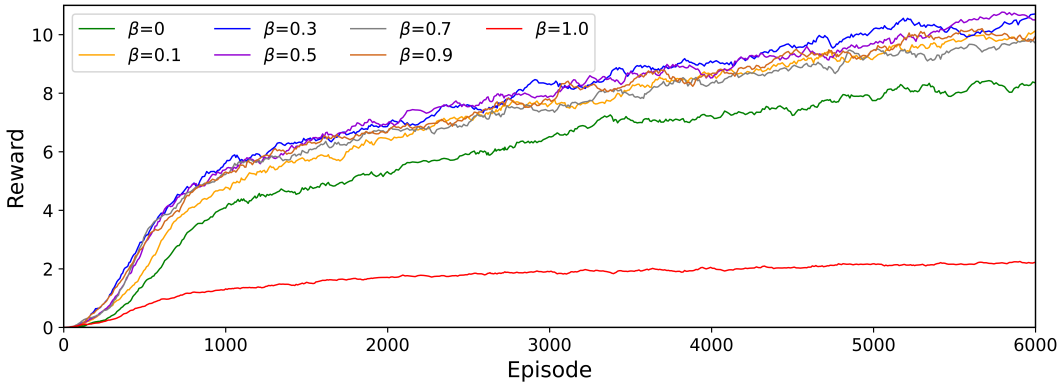
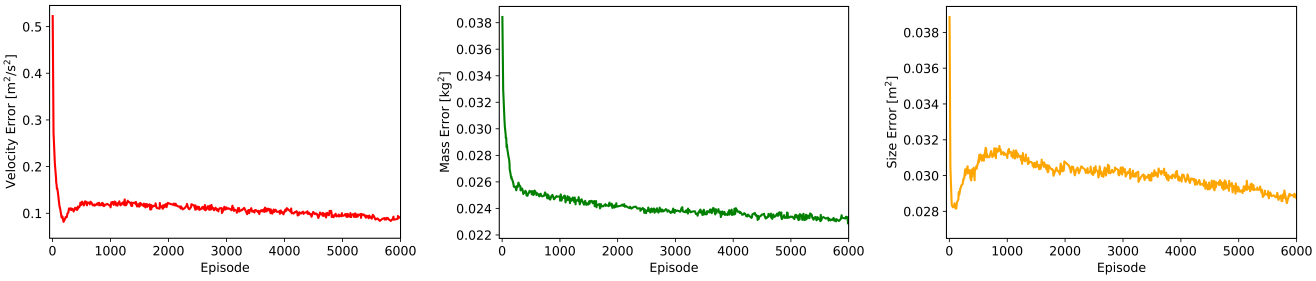


Fig. S3: The learning performance of different regression loss weight β.

In summary, when the morph net is solely guided by regression loss ($\beta = 1.0$), it proves to be significantly inadequate for learning or enhancing performance. In this circumstance, the morph net only focuses on making more precise predictions regarding the morphological constant and body velocity, while is not guided toward the direction that is beneficial to the rough-terrain locomotion tasks. When influenced only by policy gradients ($\beta = 0$), we still observe a consistent improvement in performance over time, albeit not reaching optimal levels. Interestingly, when subjected to the influence of both types of loss simultaneously, these controllers exhibit superior performance, especially when the regression loss weight β is set to 0.3 or 0.5.

We also provide a clearer elaboration of the regression errors, as shown in Fig. S4. It presents the different regression errors of morph net throughout 6000 training episodes, including velocity, mass, and size errors. The velocity error decreases sharply within the first few hundred episodes and then plateaus, maintaining a value around 0.1 for the remainder of the episodes. This indicates that the morph net quickly learned to predict velocity with great accuracy and maintained this performance consistently throughout training. The mass error converges to around 0.024 and exhibits stability, implying that the network has effectively learned to predict mass with precision after an initial period of adjustment. The size error fluctuates slightly more than the mass error but generally maintains within a consistent range and converges to around 0.028. Overall, morph net demonstrates accuracy in predicting velocity, mass, and size.



(a) Velocity Error (b) Mass Error (c) Size Error

Fig. S4: The different regression errors of MorAL.

III. Comparison Between Estimated and Ground Truth Morphology

To investigate the efficacy of the morph net estimation compared with the ground-truth constants, two kinds of input to the policy net in terms of morphology are compared 1) $[\hat{\boldsymbol{o}}_t^{mor}]$ (ours); 2) $[\boldsymbol{o}_t^{mor}]$ (ground-truth, which can be obtained from URDF files). As for 2), the input to the policy net is replaced with $[\boldsymbol{o}_t, \hat{\boldsymbol{v}}_t, \boldsymbol{o}_t^{mor}]$. The following experiments and metrics are designed to evaluate the performance of these two controllers:

1) The success rate of the robot traversing terrains is recorded, considering varying terrain heights.
2) For a fixed terrain, measure the time it takes for the robot to traverse a trajectory of fixed length with a specific terrain in the middle. The experiments are conducted in 2 different types of terrains: step and rough slope (See Fig. S5 for instance).

To showcase the adaptability of our control policy, we conduct the same set of analyses across three distinct robot types: HyQ, Laikago, and mini-Cheetah.

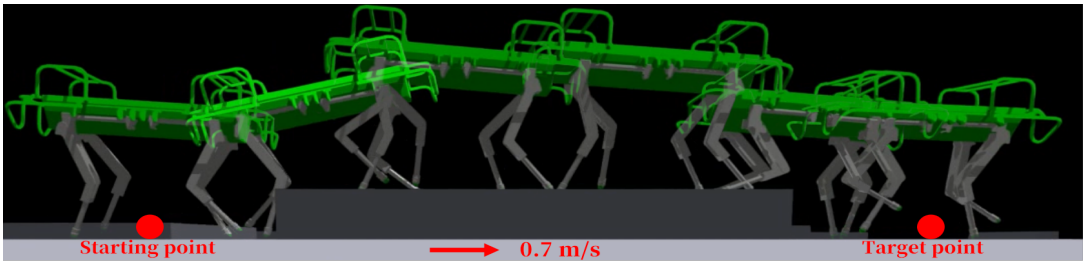


Fig. S5: Each robot is commanded to move forward at constant velocity ($0.7m/s$) and traverse a 5.17-meter-long trajectory, featuring a fixed terrain in the middle. The total time taken to traverse from the starting point to the target point is recorded (taking HyQ for example).

In the first experiment, the robot receives a forward command of $0.7m/s$. A test is deemed successful if the robot climbs up the step with both front and hind legs. We perform 20 tests for each step height and calculate the success rate. As depicted in Fig. S6, our method outperforms the policy that is trained with the ground truth morphology, which enables the robot to successfully traverse all steps approaching its mechanical limit.

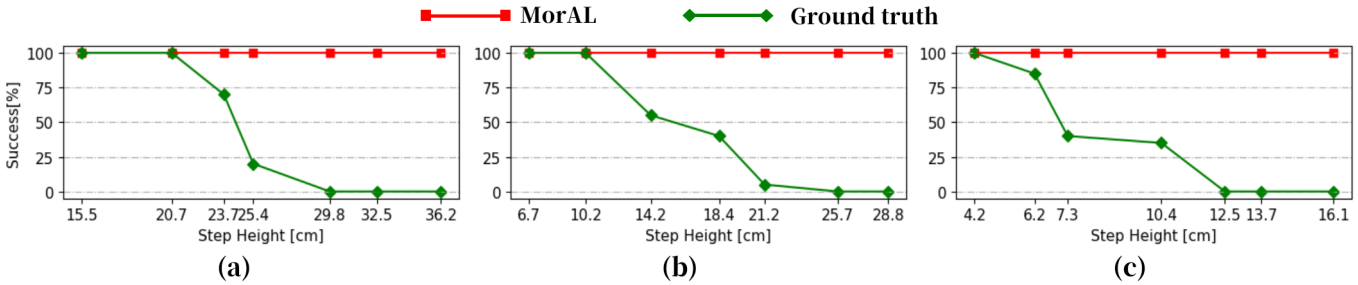


Fig. S6: Success rates comparison between MorAL $\hat{\boldsymbol{o}}_t^{mor}$ and ground truth $\boldsymbol{o}^{mor}$ over HyQ (a), Laikago (b), and Mini Cheetah (c).

Furthermore, as can be seen in Table I, the policy using the morph net estimation demonstrates a shorter traversal time across uneven terrain. The performance differences shown in these two tests are likely attributed to richer information embedded in our policy. As outlined in the loss in Equ. 4 in the article, during the training of our policy, both the gradient of the regression loss and the PPO loss (policy loss) backpropagate to the morph net. The morph net is also updated toward the direction that is beneficial to rough terrain locomotion across various robot models. In contrast, the ground-truth link masses and lengths are constant and only contain part of the information of morph net estimation. This result can also be supported by Fig. 7 in the main article that the $\hat{\boldsymbol{o}}_t^{mor}$ contains information in terms of configuration beyond just the link's length and mass.

TABLE I: The difference between traversing time of $\hat{\boldsymbol{o}}_t^{mor}$ and $\boldsymbol{o}^{mor}$.

Robot models	**Step terrain**		**Rough slope terrain**	
	MorAL	Ground truth	MorAL	Ground truth
Laikago	7.52	9.22	8.64	10.78
HyQ	7.18	7.74	7.56	9.20
Mini-Cheetah	9.54	10.66	11.20	14.38

IV. Evaluation on Actuator Net

The actuator network plays a critical role in addressing the sim-to-real transfer in this pipeline. The ablated analysis is emphatically carried out between the control policy trained with and without the actuator network in the following:

Firstly, we compare the response curves between the real actuator measurement, the actuation model without the actuator network, and the actuator network trained with real actuator measurement, as shown in Fig. S7. Here the actuation model without the actuator network is ideal, which assumes that no communication delay, instantly generating any commanded torque (zero latency).

It can be seen that the torque predicted by the actuator net coincides well with the estimated torque from the real actuator, whereas that of the model without the actuator network deviates a lot. The actuator net also accurately predicts the impulsive nonlinear torque behavior when the robot executes a highly dynamic task (Fig. S7 (**Right**)). When benchmarked against real

measurements, the learned model exhibits a significantly lower average prediction error (0.41 Nm) compared to the ideal model (4.43 Nm). In our training framework, the actuator network (blue dotted line) is employed so that the robot captures the real actuator nonlinear properties in the simulator.

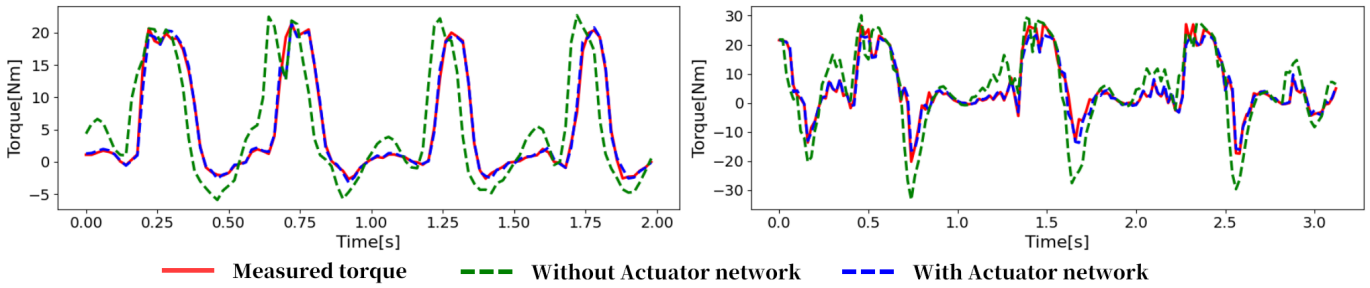


Fig. S7: The actuator response when the robot executes moderate task – trotting (**Left**) and high dynamic task – jumping (**Right**)

Next, these two kinds of actuator models are used for policy training for the same epochs. The controller trained without the actuator net induces instability, preventing the robot from maintaining a consistent motion without falling, as shown in the supplemented video and Fig. S8. Intense oscillations occur in the joints, legs, and body (Fig. S8 (**Right**)), possibly attributed to inadequate consideration of various delays (Fig. S7). Moreover, the accumulated oscillation easily leads to the abrupt change of the joint position surpassing the safety limit and triggering the machine's shutdown due to internal protection. On the contrary, modeling based on neural networks is capable of exploiting the full capacity of the machine. It enables the robot to successfully accomplish versatile tasks, conquering uneven terrain from omni-directions (Fig. S8 (**Left**)).

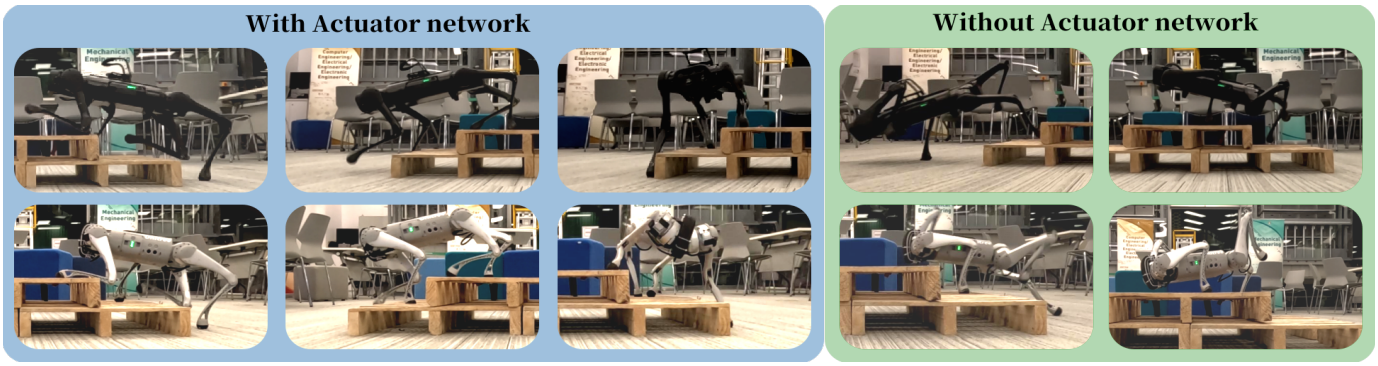


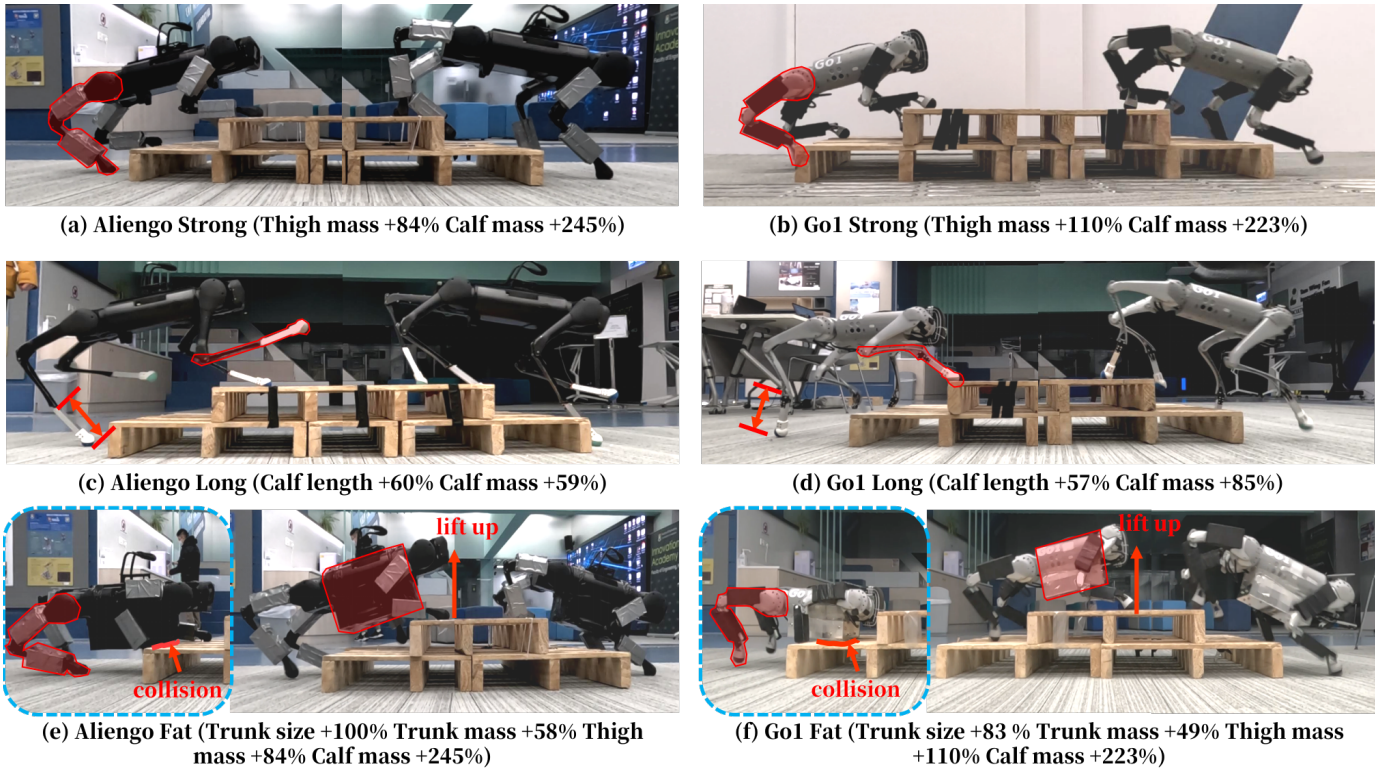
Fig. S8: The controller trained with network-predicted torque (**Left**) exhibits much more stable and safer motion compared to the controller trained with ideal torque (**Right**). The validation experiments are repeated on two types of actuators.

V. Generalization across Morphologies

This section supplements the experiment of validating the controller's generalization across morphologies. A series of robots with distinct lengths, weights, and sizes are tested, with the details of distinctive parameters presented in Table II. The full results are shown in Fig. S9.

TABLE II: The difference between different types of real-world robots.

Robot	**Mass** $[kg]$				**Link size** $[m]$		
	Total	Trunk	Thigh	Calf	Trunk (x, y, z)	Thigh	Calf
Go1	12.05	5.20	0.39	0.13	(0.39, 0.26, 0.09)	0.21	0.21
Go1 Strong	14.93	5.20	0.82	0.42	(0.39, 0.26, 0.09)	0.21	0.21
Go1 Long	12.49	5.20	0.39	0.24	(0.39, 0.26, 0.09)	0.21	0.33
Go1 Fat	17.46	7.73	0.82	0.42	(0.39, 0.26, 0.17)	0.21	0.21
Aliengo	22.90	9.04	0.64	0.22	(0.66, 0.30, 0.11)	0.25	0.25
Aliengo Strong	27.22	9.04	1.18	0.76	(0.66, 0.30, 0.11)	0.25	0.25
Aliengo Long	23.42	9.04	0.64	0.35	(0.66, 0.30, 0.11)	0.25	0.40
Aliengo Fat	32.47	14.29	1.18	0.76	(0.66, 0.30, 0.22)	0.25	0.25



(a) Aliengo Strong (Thigh mass +84% Calf mass +245%)

(b) Go1 Strong (Thigh mass +110% Calf mass +223%)

(c) Aliengo Long (Calf length +60% Calf mass +59%)

(d) Go1 Long (Calf length +57% Calf mass +85%)

(e) Aliengo Fat (Trunk size +100% Trunk mass +58% Thigh mass +84% Calf mass +245%)

(f) Go1 Fat (Trunk size +83 % Trunk mass +49% Thigh mass +110% Calf mass +223%)

Fig. S9: The robot morphology varies in a wide range, while the controller implicitly identifies those differences via proprioception and adjusts the action accordingly.

Besides, to verify the controller's adaptability to the dynamic morphology change, the weights of the robot's legs and base (body) are dynamically varied during the robot locomotion process.

In this test, we arbitrarily attach the payload to the robot while it traverses the terrain, as showcased in Fig. S10 and the accompanying videos. The weight of the robot's legs and base experience sudden increases, and the base and thigh limb weight can increase up to 57% and 156% of its original ones, respectively. As can be seen in Fig.S10(b), even though the suddenly heavier legs cause the robot to lean toward the heavier side, the robots quickly adjust the action and maintain balance.

(a) (b)

Fig. S10: The body mass (a) and limb mass (b) are dynamically changed during the robot locomoting on the rough terrain.

VI. Complementary Details

A. Details on Ablation Study Setup

By excluding one or more estimated states from the output of morph net, we can obtain four new controllers, including w/o size estimator, w/o mass estimator, w/o size and mass estimator, and w/o all estimator. When training these new controllers, we directly remove the corresponding output dimension of the morph net. The outputs and the dimensions of different morph net are listed in Table III to provide a clearer comparison of the five ablated controllers, as follows:

TABLE III: The output and dimension of five controllers with different morph parameters.

Controller	Output of the Morph Net	Dimension
MorAL	$[\hat{\boldsymbol{v}}_t,\ \hat{\boldsymbol{m}}_t, \hat{\boldsymbol{s}}_t]$	12
w/o all estimator	None	0
w/o size estimator	$[\hat{\boldsymbol{v}}_t,\ \hat{\boldsymbol{m}}_t]$	7
w/o mass estimator	$[\hat{\boldsymbol{v}}_t,\ \hat{\boldsymbol{s}}_t]$	8
w/o mass and size estimator	$[\hat{\boldsymbol{v}}_t]$	3

where $\hat{\boldsymbol{v}}_t$, $\hat{\boldsymbol{m}}_t$, and $\hat{\boldsymbol{s}}_t$ represent the estimated body velocity, the estimated masses of trunk and legs, and the estimated sizes of trunk and legs, respectively.

B. Details on Network Update Mechanism

As shown in Fig. S11, the output of the morph net $[\hat{\boldsymbol{v}}_t, \hat{\boldsymbol{o}}_t^{mor}]$ serves as the input of the policy net, as indicated by the pink line. Therefore, the gradient of $loss_{policy}$ can be backpropagated not only to the policy net but also backpropagated to the morph net, as indicated by the red dashed line. The morph net is not only updated in a regression fashion with the ground truth morphological parameters ($loss_{reg}$) but also updated by the RL aims to improve locomotion task performance ($loss_{policy}$). Since different morphological robot models have different policy gradient directions to improve their task performance, as the training converges, the morph net is updated towards their respective directions.

For the following reasons, the hidden state $\hat{\boldsymbol{o}}_t^{mor}$ carries knowledge about the configuration and can infer the configuration difference between FKBE, FRBK, FE, and FK from the robot state history. To further validate the following reasons, we train a classification net $\boldsymbol{E}_{\mathcal{C}}$ to predict the category of configuration $\hat{\mathcal{C}} \in \{0, 1, 2, 3\}$ using the latent state $\hat{\boldsymbol{o}}_t^{mor}$, where 0, 1, 2, 3 represent category of FKBE, FRBK, FE, and FK, respectively. The $\boldsymbol{E}_{\mathcal{C}}$ is updated to minimize the cross-entropy loss calculated with the predicted probability vector and the ground-truth category. Lastly, we visualize predicted probability vectors on a test set containing a variety of robot configurations, as shown in Fig. S12. It can be seen that $\hat{\mathcal{C}}$ can be classified due to the encoded knowledge from $\hat{\boldsymbol{o}}_t^{mor}$.

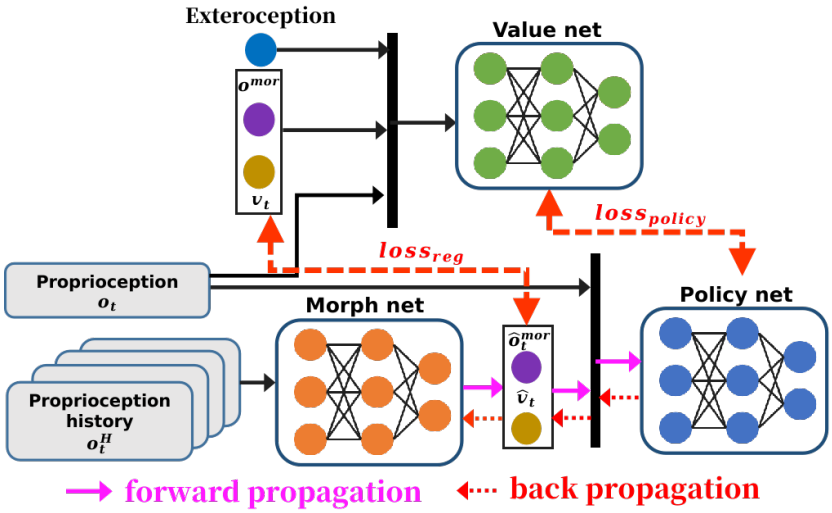


Fig. S11: The flowchart of the update process for morph net.

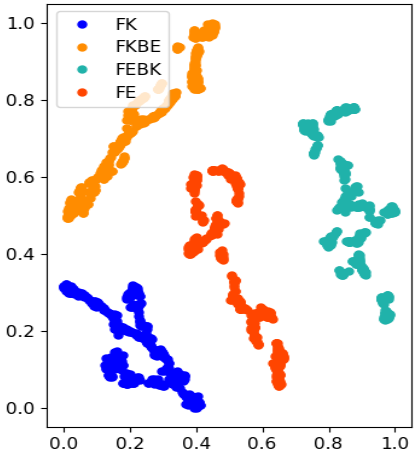


Fig. S12: $\hat{\boldsymbol{o}}_t^{mor}$ can be implicitly identified and classified as different configurations