

# Identifying Terrain Physical Parameters From Vision - Towards Physical-Parameter-Aware Locomotion and Navigation

Jiaqi Chen<sup>1</sup>, Graduate Student Member, IEEE, Jonas Frey<sup>2</sup>, Graduate Student Member, IEEE, Ruyi Zhou<sup>3</sup>, Takahiro Miki<sup>4</sup>, Graduate Student Member, IEEE, Georg Martius<sup>5</sup>, and Marco Hutter<sup>6</sup>, Member, IEEE

**Abstract**—Identifying the physical properties of the surrounding environment is essential for robotic locomotion and navigation to deal with non-geometric hazards, such as slippery and deformable terrains. It would be of great benefit for robots to anticipate these extreme physical properties before contact; however, estimating environmental physical parameters from vision is still an open challenge. Animals can achieve this by using their prior experience and knowledge of what they have seen and how it felt. In this work, we propose a cross-modal self-supervised learning framework for vision-based environmental physical parameter estimation, which paves the way for future physical-property-aware locomotion and navigation. We bridge the gap between existing policies trained in simulation and identification of physical terrain parameters from vision. We propose to train a physical decoder in simulation to predict friction and stiffness from multi-modal input. The trained network allows the labeling of real-world images with physical parameters in a self-supervised manner to further train a visual network during deployment, which can densely predict the friction and stiffness from image data. We validate our physical decoder in simulation and the real world using a quadruped ANYmal robot, outperforming an existing baseline method. We show that our visual network can predict the physical properties in indoor and outdoor experiments while allowing fast adaptation to new environments.

**Index Terms**—Legged Robots, deep learning for visual perception, field robots.

Received 30 March 2024; accepted 15 August 2024. Date of publication 6 September 2024; date of current version 19 September 2024. This article was recommended for publication by Associate Editor S. Saripalli and Editor A. Valada upon evaluation of the reviewers' comments. This work was supported in part by the Max Planck ETH Center for Learning Systems (Frey), in part by Swiss National Science Foundation (SNSF) under Project 166232 and Project 188596, in part by the European Union's Horizon 2020 Research and Innovation Program under Grant 101016970 and Grant 101070405, and in part by National Natural Science Foundation of China Youth Basic Research Project (Doctoral Student) under Grant 523B2039. (Corresponding author: Jonas Frey.)

Jiaqi Chen, Takahiro Miki, and Marco Hutter are with the Robotic Systems Laboratory, ETH Zurich, 8092 Zurich, Switzerland (e-mail: chenjiaq@ethz.ch; tamiki@ethz.ch; mahutter@ethz.ch).

Jonas Frey is with Robotic Systems Laboratory, ETH Zurich, 8092 Zurich, Switzerland, and also with the Max Planck Institute for Intelligent Systems Tübingen, 72076 Tübingen, Germany (e-mail: jonfrey@ethz.ch).

Ruyi Zhou is with the Robotic Systems Laboratory, ETH Zurich, 8092 Zurich, Switzerland, and also with the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin 150080, China (e-mail: zhouryh@hust.edu.cn).

Georg Martius is with the Max Planck Institute for Intelligent Systems Tübingen, 72076 Tübingen, Germany, and also with the University of Tübingen, 72076 Tübingen, Germany (e-mail: georg.martius@tuebingen.mpg.de).

Project Page <https://bit.ly/3Xo5AA8>

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2024.3455788>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3455788

## I. INTRODUCTION

LEGGED robots excel in walking on challenging terrains [1], [2], [3], [4], offering advantages in search and rescue [5], [6], planetary exploration [7], [8], and hazardous area navigation [9]. The main challenge lies in ensuring their robustness and adaptability to diverse ground conditions like slippery surfaces, soft soils, and uneven terrains, necessitating control and navigation systems capable of sensing and responding to the environmental variations.

Physical simulators [10], [11] are vital for training locomotion [1], [3] and navigation [12] policies for legged robots using reinforcement learning. They enhance controller robustness by simulating diverse terrains and conditions like rough or slippery surfaces [3], [13], [14], [15] and varying ground stiffness [16], [17], enabling robots to adapt to different environments. While randomizing physical parameters can improve the locomotion robustness, these parameters only become observable through direct interaction with the terrain [1], [3]. It would be more beneficial for robots to anticipate conditions such as slippery surfaces or foot sinkage using exteroception before contact. However, learning this in simulation is difficult. Although we can create photorealistic images, accurately replicating the physical effects of interacting with terrain in varied stiffness and friction is challenging. For example, simulating realistic grass visually is possible, but it's hard to match its real-world physical properties perfectly. This mismatch between simulated and real-world terrain properties poses a challenge in transferring policies trained in simulation to the real world.

Self-supervised learning approaches allow training models to predict terrain-related properties from real-world images by labeling images from different sources [18], [19], [20], [21], [22], [23]. Recent approaches predict terrain-related property representations from vision, such as traversability [18], [19], empirical ground reaction scores [22], or learned audio embeddings [23]. These predictions are supervised using metrics like velocity tracking error, foot-terrain interaction force/torque, or vehicle-terrain interaction sounds. Despite these models can predict various properties visually, they cannot be directly used for training locomotion or navigation policies in simulation due to the challenge of accurately simulating signals like foot-terrain interaction force/torque or sound.

In this work, we propose a framework (Fig. 1) that aims to predict simulation parameters, instead of metrics like traversability, from vision using self-supervised learning. These simulated parameters are defined in simulation and don't exactly match real-world physical parameters. For example, stiffness = 0.5 in

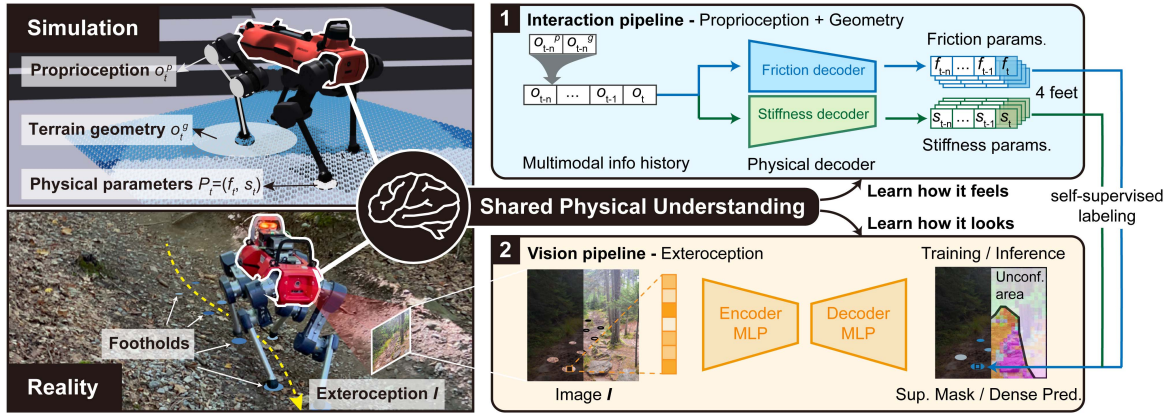


Fig. 1. Overview of the two-stage self-supervised terrain physical parameter learning framework. A physical decoder in twin structure is trained in simulation to predict simulated friction and stiffness parameters per foot. The physical decoder transfers to the real world, where it provides self-supervised labels (within the supervision mask) to train a visual network on real-world image data. In the training stage, the visual network is trained with weak supervision only on the foothold pixels. In the inference phase, the visual pipeline processes all pixel features within an image and outputs the corresponding dense prediction of the simulated parameters with a confidence mask.

simulation does not equal a stiffness coefficient  $k = 0.5$  in the real world. By connecting real-world images with their corresponding physical properties in simulation, we can utilize these estimations for policy-training in simulation. The framework consists of two stages: In stage one, a *physical decoder* is trained in simulation to estimate physical parameters of the terrain from robotic proprioception and geometric sensing of the terrain. Using the simulated robotic interactions with different terrains, the physical decoder learns to estimate the simulated friction and stiffness that best approximate the behavior on various terrains. In stage two, a *visual network* is trained with real-world data to map visual features to simulated physical parameters using labels generated by the physical decoder. The vision pipeline also uses anomaly detection to assess the reliability of the predicted terrain properties. A continuous learning process ensures that the robot can adapt to the complexity and uncertainty of real-world terrains, due to the fact that how it looks does not uniquely determine how it feels.

The main contributions of our work are as follows:

- *Proposing a physical properties learning framework* aiming at transferring physical-parameter-aware locomotion and navigation policy to the real world.
- *The physical decoder*, a recurrent neural network with a gating mechanism trained in a contact-enhanced simulation, is able to predict terrain friction and stiffness parameters per foothold.
- *The visual network*, trained in a self-supervised manner to predict the simulated terrain parameters from the images, can be continuously updated and adapts quickly to new scenarios during real-world deployment.
- *Experimental analysis and quantitative evaluation* in simulation and real world demonstrating the superior performance of our physical decoder over the baseline method, as well as the efficacy of the visual pipeline.

## II. RELATED WORK

### A. Estimating Terrain Properties From Interaction

Many studies have already shown that legged robots can efficiently estimate terrain-related properties through

interaction. Xu et al. [16] estimate friction and stiffness with sensor-measured normal and tangential contact forces acquired in formulaic pressing and rubbing motions for hexapod robots. Yu et al. [24] explore the prediction of terrain roughness using hall effect sensors and a novel whisker-based system. Margolis et al. [25] predict friction and roughness using a neural network with an active-sensing policy incorporating probing motions. These prior works either require additional sensors or rely on probing motions which hinder the robot from moving freely. Miki et al. [3] train a recurrent belief state decoder in simulation, which is a part of the locomotion policy, to predict foot contact information, including friction coefficient. The belief decoder is implemented by a recurrent architecture, which processes historical sensor data, and shows its capability to estimate friction when the robot steps onto a slippery platform without relying on any specific motions or sensors. Inspired by this design, we train a network to predict friction and stiffness accurately for each foot.

### B. Estimating Terrain Properties From Vision

Identifying terrain properties through interaction is well studied, but poses potential risks, such as being slippery or stuck due to excessive foot sinkage. In contrast, vision-based methods offer the opportunity to estimate these properties from semantic-rich images and help avoid these hazards in advance. Multiple works explore extracting meaningful feature embeddings from image data [26], [27]. Correlating semantic information to those feature embeddings relying on pre-trained models has been shown to be specifically effective with limited available training data [18], [23], [25].

A common approach to obtain labels for image training is to use self-supervision in hindsight and cross-modality [18], [19], [22], [23]. Wellhausen et al. [22] train a Convolutional Neural Network (CNN) to associate a learned terrain property metric named *ground reaction score* with camera images. Margolis et al. [25] used the predicted values from a physical estimator as labels to train a linear layer on the dense feature representation. Lee et al. [28] reproject footholds to learn the geometry of the support surface from image data. Frey and Mattamala et al. [18], [19] reproject the footprint of the robot labeled with the velocity

tracking error to approximate traversability and use it as a label in the image space. In this work, we use a similar self-supervision principle with a label-projection method as [22], [28], adapting per-foot labeling.

Even though pre-trained features offer good generalization, models trained on limited data may not adapt to entirely new environments. Following [18], [19], we adapt the network to the deployment environment online during the mission and predict the physical parameters of the environment, allowing to transfer physical-parameter-aware locomotion and navigation policies to the real world.

### C. Anomaly Detection

Anomaly detection methods enable identification of the input data that is Out-Of-Distribution (OOD). Most approaches model the input data distribution and use a similarity measure to determine if a sample is OOD, based on a predefined threshold. Richter and Roy [29] train an autoencoder to reconstruct image data and use the reconstruction loss threshold to identify OOD samples. Wellhausen et al. [9] use a loss threshold with a normalizing flow network. Frey et al. [18] follow [29] and dynamically adjust the anomaly detection threshold by fitting a Gaussian distribution to positive samples. We extend [18], however, fit a Gaussian Mixture Model (GMM) with two components that yield a hyperparameter-free threshold.

## III. METHOD

The framework (Fig. 1) for online and self-supervised physical environment understanding consists of two stages. The physical decoder provides terrain parameter labels for the online training of the visual network; The visual pipeline outputs confidence-masked dense predictions of the physical properties of the environment in image space.

The *physical decoder* (Section III-A) has a twin network to estimate the friction and stiffness values of each foothold, respectively. These footholds, along with their associated friction and stiffness values, are then projected into the camera image, where they serve as labels for training the visual network. The visual network uses a Multi-Layer Perceptron (MLP) to predict the physical parameters (Section III-B) from pixel-wise feature embeddings, and a confidence mask is generated using anomaly detection (Section III-C). A *Mission Graph* and a *learning thread* (Section III-D) are established for online dataset storage and training of the visual network.

### A. Physical Decoder

As illustrated in Fig. 1, we train a physical decoder capable of predicting both terrain friction and stiffness on a per-foot basis using proprioceptive and exteroceptive information within a simulation environment. The decoder learns the correspondence between the simulated terrain parameters and the resulting motion in the simulation, which is then intended to be transferred to real-world applications. The decoder consists of a twin network architecture, as demonstrated in Fig. 2. Both networks share the same input observation history  $O = \{o_{t-n}, \dots, o_t\}$ , where  $n$  is the history length. Each observation  $o_t$  comprises the robotic proprioception  $o_t^p$ , including command, joint, body information as well as leg phase information, along with the terrain geometry  $o_t^g$  represented by height samples around each foot in a circular sampling pattern, as [3]. Each decoder predicts

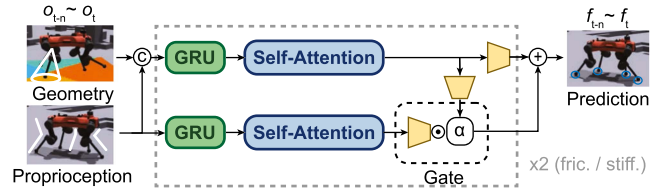


Fig. 2. Physical decoder architecture in the form of a twin network. Friction and stiffness are predicted by each separate network. The yellow trapezoidal blocks are MLPs.

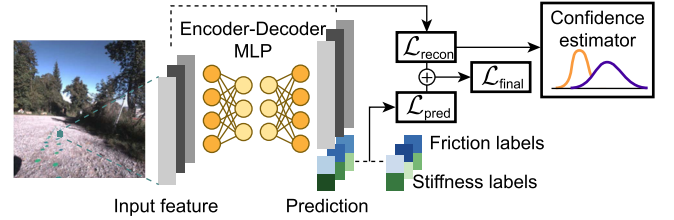


Fig. 3. Visual network architecture and losses used for training. The decoder is in an encoder-decoder structure for the simultaneous OOD detection and physical parameters regression. Friction and stiffness values of each pixel feature in the input image are predicted at the same time.

a 4-dimensional vector representing the friction or stiffness per foothold. When the foot is in the swing phase, the corresponding friction or stiffness is determined by the terrain parameter within the foot's projection area on the ground. Inspired by the recurrent structures in [3], each twin network comprises Gated Recurrent Units (GRUs) blocks, self-attention layers, and MLP prediction heads with an additional gating mechanism. The GRUs retain relevant historical information for friction or stiffness estimation, which is further refined by the self-attention layers. Considering two different modalities of input information, the network architecture consists of a proprioception-only and concatenated-information path. In each path, a MLP predicts the physical terrain properties. The gating mechanism allows to weight the contribution of each path. Unlike [3], where terrain property decoding is treated as an auxiliary task, we optimize the physical decoder independently from the locomotion policy. Additional details on the network architecture, training environment, data collection, and training hyperparameters are provided in Section III-E.

### B. Visual Network

We implement the visual network in an encoder-decoder structure (Fig. 3). We first extract pixel-wise feature embeddings using a pre-trained DINOv2 backbone [27], compared to DINOv1 [26] used in [18]. Since friction and stiffness aren't strongly dependent on terrain geometry, using powerful image features extracted by DINOv2 containing implicit geometric and other information is enough to build a good mapping between these parameters and features without explicit depth sensing. In addition, we follow [19], an extension of [18], and predict the terrain properties per-pixel instead of using SLIC superpixels [30] to aggregate feature embeddings.

The visual network outputs the reconstructed input feature for anomaly detection (Section III-C) in confidence estimator and friction and stiffness predictions per pixel. The total loss per



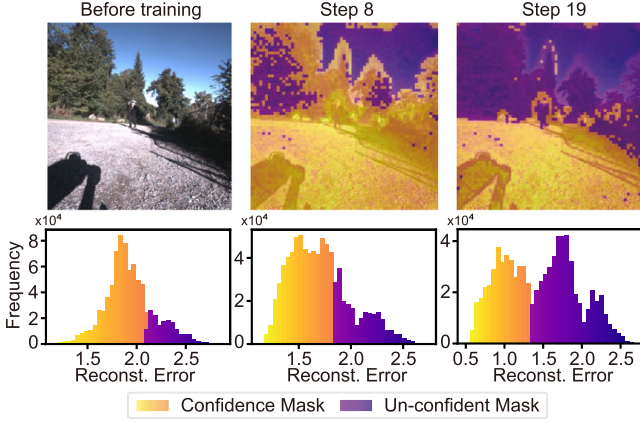


Fig. 4. Evolution of the reconstruction loss distribution with the increase of training steps. The reconstruction loss distribution is unimodal before training, while changed to a bimodal distribution during training. Yellow-orange indicates In-Distribution (ID) data, while purple-blue is for Out-Of-Distribution (OOD) data.

pixel during training is a weighted combination of reconstruction loss  $\mathcal{L}_{\text{recon}}$  and regression loss  $\mathcal{L}_{\text{pred}}$  given by:

$$\mathcal{L}_{\text{recon}} = \text{MSE}(x_{\text{recon}}, x_{\text{label}}) \quad (1)$$

$$\mathcal{L}_{\text{pred}} = \text{MSE}(y_{\text{pred}}, y_{\text{label}}) \quad (2)$$

$$\mathcal{L}_{\text{final}} = w_{\text{recon}} \cdot \mathcal{L}_{\text{recon}} + w_{\text{pred}} \cdot \mathcal{L}_{\text{pred}}, \quad (3)$$

where  $w_{\text{recon}}$  and  $w_{\text{pred}}$  are the weighting coefficients for the reconstruction and prediction loss respectively. More implementation details on the label generation for the visual network training, and the underlying graph storage implementation are provided in Section III-D.

### C. Anomaly Detection

When deploying legged robots, they interact only with a small part of the scene. Our self-supervised labeling method, therefore, can only provide physical parameter supervision for regions with footholds. However, the visual network predicts parameters for the full image including irrelevant areas (e.g. sky and trees). This directly necessitates reasoning about the reliability of these predictions. We implement an anomaly detection strategy using the encoder-decoder architecture of the visual network to predict OOD regions, by learning to reconstruct the feature embedding of pixels with associated physical properties. Following the implementation of [18], we observed a bimodal reconstruction error distribution during training as shown in Fig. 4.

To dynamically adjust the threshold for OOD discrimination, we use a Gaussian Mixture Model (GMM) [31] with  $k = 2$  and *full* covariance type to fit the bimodal distribution. Fitting a GMM to the histogram is independent of the training or inference process of the visual network and the threshold can be dynamically adapted per image. This process requires no hyper-parameter tuning across datasets and we will show its advantage in Section IV-C.

### D. Online Training

The visual network is trained online during the deployment to overcome the generalization limitations of offline training on limited datasets. It can update the mapping between visual

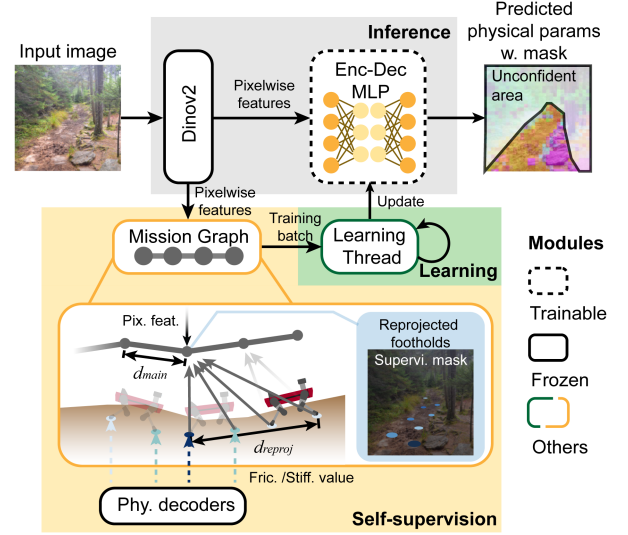


Fig. 5. Online training framework adapted from [18]. The inference task extracts features and outputs masked dense predictions. The self-supervision task contains the Mission Graph to store paired input features and labels provided by the physical decoder. The learning task performs continuous training of the visual network in the learning thread.

features and physical parameters after it encounters a different pattern during environmental interaction. We perform three tasks in parallel during the mission, as shown in Fig. 5. *Task 1*: fast prediction of terrain properties based on the received image; *Task 2*: self-supervised labeling of training data by associating footholds (only when in contact) with physical terrain parameters to camera images; *Task 3*: training the visual network given the labeled data. All three tasks are connected by the Mission Graph, the central hub, allowing to receive, store, and transfer training-relevant data.

1) *Mission Graph*: The Mission Graph accumulates training data by storing image features and the associated physical parameters. To facilitate associating image features and terrain parameters, each mission node stores the dense feature map, a supervision mask of footholds with associated friction and stiffness, of the same height and width as the camera image, and the associated camera intrinsics and extrinsics.

2) *Inference Task*: This task processes camera images by extracting dense features using DINOv2. The visual network predicts friction and stiffness values using the dense pixel-wise features and outputs a confidence mask for each image.

3) *Self-Supervision Task*: The self-supervision task receives the generated dense features from the inference task with the associated camera intrinsic and extrinsic, as well as the labeled footholds with physical parameters from the physical decoder. Dense features are stored in a new node within the Mission Graph if and only if the camera image is recorded at a position farther away than  $d_{\text{main}}$ , with respect to the latest added node. When a new foothold with labeled physical parameters is received, the foothold is projected on all camera images stored within a range of  $d_{\text{reproj}}$  in the Mission Graph using the same projection method in [22]. The projection is used to fill the supervision masks with the friction and stiffness values. This establishes the pairing of feature embeddings to terrain physical parameters.

4) *Learning Task*: In each iteration of the learning process, a batch of mission nodes containing a supervision signal are

randomly selected. The resulting training batch is used to update the visual network using gradient-based optimization.

### E. Implementation Details

1) *Interaction Pipeline*: In the physical decoder architecture, the 1-layer GRUs have a 100-dimensional hidden state. The number of hidden and output units of the prediction head MLPs is [64,32,4], while it is [64,64,4] for the MLP controlling the sigmoid gating unit. All MLPs use LeakyReLU activation functions. The history length of input observation sequences is 50, and input sequences are zero-padded for missing historical information during both training and testing. During training, the hidden states of the GRUs are reset to zeros for each new sequence.

For data collection, all the data for training and testing the physical decoder is collected in legged gym [10] enhanced with procedurally generated terrain [6] and an additive soft terrain contact model [17]. The training and testing environments are randomly generated per terrain patch with varying geometry, stiffness, and friction parameters. The friction value ranges from 0 to 1, and stiffness from 1 to 10. The legged robots are controlled by a robust locomotion policy [3] in simulation, and their command velocities are uniformly sampled between  $-1.3$  m/s and  $1.3$  m/s with an angular velocity between  $-0.3$  rad/s and  $0.3$  rad/s. We collect a dataset consisting of 18.3 h and 5 h real-time equivalent locomotion in simulation for training and testing.

For training hyper-parameters, we use a batch size of 64. Label normalization is dynamically performed by continuously updating the running mean and standard deviation when new data is ingested. Furthermore, labels are weighted by their interval-based frequencies when calculating the loss, ensuring a balanced training process. For the optimizer, we use Adam [32] with a learning rate of 0.001 and a weight decay factor of 0.00001. We train for a total of 100 epochs, which takes around 30 minutes on a single GPU.

2) *Vision Pipeline*: In the visual network's encoder-decoder architecture, hidden layers are configured as [128,32,128, 384+2], where the output of 384 corresponds to the input feature dimension, while 2 additional channels are used for the friction and stiffness prediction. In preprocessing, we first scale the camera image to a height of 1078 and then center-crop it to  $910 \times 910$  pixels. The loss weights are set to:  $w_{pred} = 0.1$  and  $w_{recon} = 0.9$ . For the Mission Graph, the default outdoor setting is  $d_{main} = 1$  m,  $d_{reproj} = 5$  m, and  $d_{min}$  is set to 0.2 m for indoor experiments. For online training, we use the Adam optimizer with a learning rate of 0.001, weight decay factor of 0.001, and batch size of 100. The training loop runs at around 2.0 Hz. For key hyperparameters, we conducted a grid search and chose the best based on the validation dataset.

## IV. EXPERIMENTS

We test the interaction-based and vision-based components of our terrain physical parameter learning framework in effectiveness and accuracy consecutively. First, we evaluate the prediction performance of the physical decoder in simulation (Section IV-A1). Second, we compare prediction results for friction and stiffness against a baseline when walking over a slippery whiteboard and a foam board in the real world (Section IV-A2). Third, we show a quantitative analysis of our

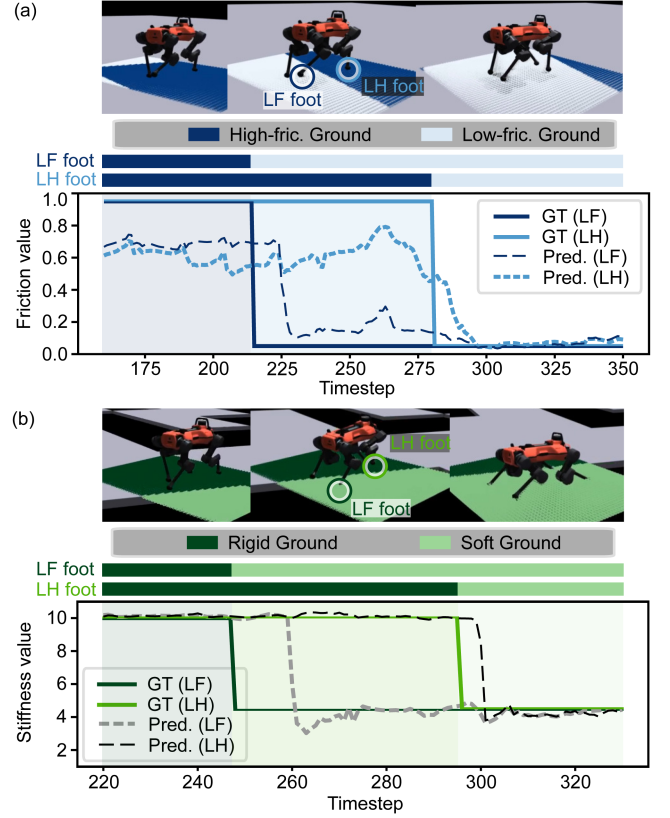


Fig. 6. Friction and stiffness estimation in simulation. One steps equals 20 ms. Prediction results shown here are from the left-front (LF) and left-rear (LH) foot.

model with a digital-twin-based experiment, where the predicted friction value in the real world results in a similar motion when rolling out the same action in simulation (Section IV-B). It justifies the usage of our physical decoder to generate labels for the self-supervised training of the visual network. Next, we evaluate our online-trained vision pipeline in an indoor environment (Section IV-C). Lastly, we compare different methods for anomaly detection. All experiments are conducted using an ANYmal D quadrupole robot with a robust locomotion policy [3] and state estimation given by [33]. Experiments were run on an Nvidia RTX 4080 GPU with Intel i7-12700H CPU.

### A. Physical Decoder

1) *Test in Simulation*: We test the effectiveness of our trained physical decoder in simulation and quantitatively evaluate its performance compared to a *Baseline* method [3] for friction estimation. As shown in Fig. 6(a), the friction prediction of the front and hind feet transits quickly and correctly over time when the robot walks from a high-friction (0.95) region to a low-friction (0.05) one. The friction prediction of the front foot drops firstly at step 225, while the decrease in friction for the hind foot happens around 60 steps later. This indicates that our friction network is able to identify parameters exactly per foot contact, rather than an average measure of four foot contacts. Within the high friction region, we can observe that the friction prediction around 0.6 doesn't perfectly match with the ground truth value of 0.95. This is due to the fact that the motion on high friction terrains (0.6 1.0) is very similar given that no foot slippage

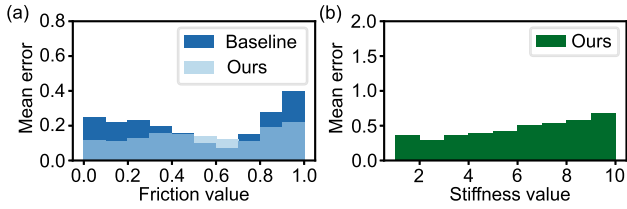


Fig. 7. Simulation test error histograms of friction (a) and stiffness (b) prediction. The Baseline is the decoder in [3].

occurs, which renders the exact friction value unobservable. Even though the prediction in high friction range is less precise, it is acceptable and sufficient to adapt the locomotion or navigation policy to varying frictions.

For the stiffness prediction, the decoder is able to estimate the terrain stiffness accurately across the full range from 1 to 10, as shown in Fig. 6(b). In both the friction and stiffness testing scenarios, we can observe a delay of around 20 timesteps (0.4 s) between the ground truth and decoder prediction. This is due to the fact that the foot enters the low friction/stiffness region but has not interacted with the new terrain patch. The statistics of the prediction error in Mean Absolute Error (MAE) histogram over the 5-hour test data in terms of friction and stiffness are illustrated in Fig. 7. Regarding friction prediction, it shows that our method outperforms *Baseline* on average in terms of MAE (0.21  $\rightarrow$  0.15). For high friction ranges, both methods cannot accurately distinguish different friction values. For stiffness prediction, our method can predict stiffness with an overall MAE of 0.46. This indicates that stiffness can be predicted precisely ( $\sim 5\%$  deviation), given the large stiffness range of 1 to 10. Notably, we observed experimentally that the gating mechanism tends to emphasize the path containing solely proprioceptive information for friction prediction, while relying more on the path containing exteroception for stiffness prediction. This observation can be attributed to the fact that stiffness is observable when the foot penetrates the ground, which can be detected based on the exteroception, whereas friction predictions are less dependent on the terrain geometry.

2) *Test in Real World*: To demonstrate that our in-simulation trained physical decoder transfers effectively to the real world, we conducted separate experiments for varying friction and stiffness. For friction, we tele-operated the robot to move backwards from a high-friction ground (*GROUND*) onto a slippery, water-covered whiteboard (*WB*). As shown in Fig. 8, our friction decoder accurately distinguishes between different friction levels per foot (e.g., rear feet on the slippery area, front feet on high-friction ground), while the *Baseline* [3] predicts the same estimate for all feet. For stiffness, we had the robot walk over a soft foam board, as illustrated in Fig. 9. Our stiffness decoder identifies the low-stiffness area when stepping on the foam and quickly adjusts to a high value upon returning to rigid ground.

## B. Digital Twin

Although the efficacy and qualitative sim-to-real transferability of the physical decoder have been demonstrated in previous experiments, the numerical correctness of the predicted simulation parameters hasn't been verified in the real world. By ensuring these parameters yield comparable motions in simulations, we can render the physical properties of the terrain observable

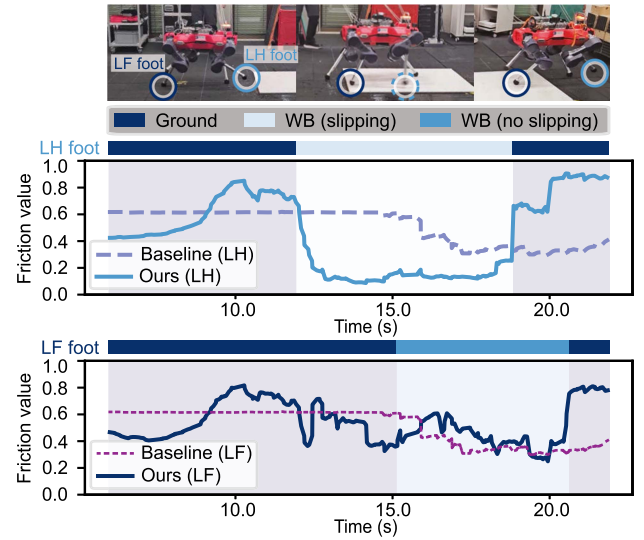


Fig. 8. Physical decoder friction estimation. Only left-front (LF) and left-rear (LH) feet are shown in the plot. The robot walks backwards.

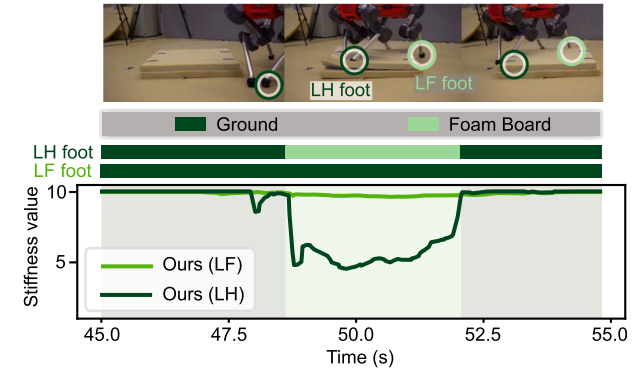


Fig. 9. Physical decoder stiffness estimation. Left-front (LF) and left-rear (LH) feet are shown in the plot.

for locomotion or navigation policies trained in simulation with privileged information.

To calculate the numerical correctness of the predicted simulation parameters in the real world, we design a digital twin experiment, inspired by [34]. It can align simulation parameters with the real world by comparing real and simulated robot motions under various physical conditions, leveraging data from the friction experiment (Section IV-A2). To quantify motion similarity, we introduce three metrics: simulated joint position error (absolute error), base orientation error (geodesic distance), and foot slippage distance error, which is calculated from the movement of the foot from ground contact to lift-off.

We initialize the simulated robot with the same base orientation, joint positions, and velocities as recorded from the real robot. Orientation roll and pitch are reliably captured by the onboard IMU, while the base height is determined by assuming one foot is always on the ground. The simulation terrain is reconstructed to match the terrain geometry in the corresponding real-world experiment. An accurate reconstruction is the key to provide discernible comparison results. Due to the current limitation of the simulation and the manual reconstruction pipeline, this method currently supports only flat, rigid surfaces. The recorded dataset is split into snippets: whiteboard (*WB*) and



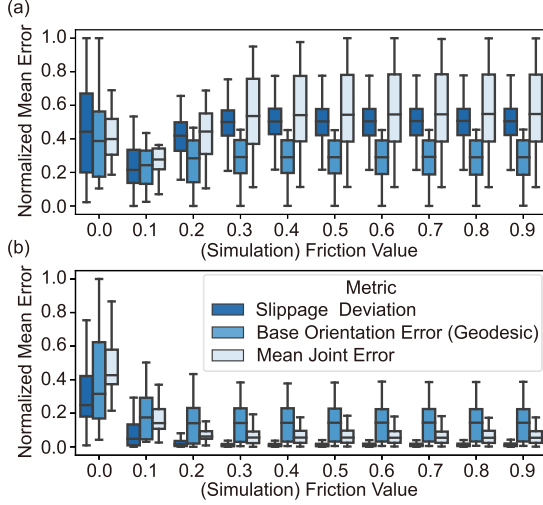


Fig. 10. Boxplot result of the digital twin experiment. (a) Error statistics of the WB experiment. (b) Error statistics of the GROUND experiment.

TABLE I  
MEAN FRICTION PREDICTION ERROR (LH FOOT)

Scene	Baseline	Ours
WB	$0.33 \pm 0.11$	$0.02 \pm 0.08$
GROUND	$0.00 \pm 0.00$	$0.00 \pm 0.00$

high-friction ground (*GROUND*). We selected 18 *WB* snippets, each 20 timestamps (400 ms) long, where foot slippage occurred, and 220 *GROUND* snippets of the same length with no slippage. The lower number of *WB* samples is due to the rarity of slippage events. We then simulate 400 ms rollouts for 10 friction parameters ranging from 0.0 to 0.9, noting that longer trajectories tend to diverge from reality. All metrics are averaged per snippet.

The results shown in Fig. 10 indicate that the lowest error range for *WB* is within a range of (0.0, 0.2) and a range of (0.3, 1.0) for *GROUND*. This result allows us to reinterpret Fig. 8, where we can observe high friction predictions ( $> 0.4$ ) on *GROUND* and low friction estimates ( $\approx 0.1$ ) on *WB*. We use these two ranges above for quantitative analysis of the LH foot in Fig. 8. The prediction error per timestep is set to zero if the predicted value is within the range. Otherwise, the error is given by the absolute error to the nearest boundary of the prediction range. The quantitative result in Table I shows that our approach outperforms the *Baseline*, particularly with *WB*, and performs comparably on *GROUND* due to its broad valid range. The results indicate that the physical decoder trained in simulation can successfully transfer to the real world under controlled conditions. Given the design of our simulation environment to closely mimic real-world physical interactions, we assume the shown results also generalize to stiffness. However further validation is still needed after we enhance the digital twin method in the future.

### C. Visual Network

We train the visual network in a self-supervised manner on similar robot data as the one in the digital twin experiment (Section IV-B). The training data consists of 22 images. Within

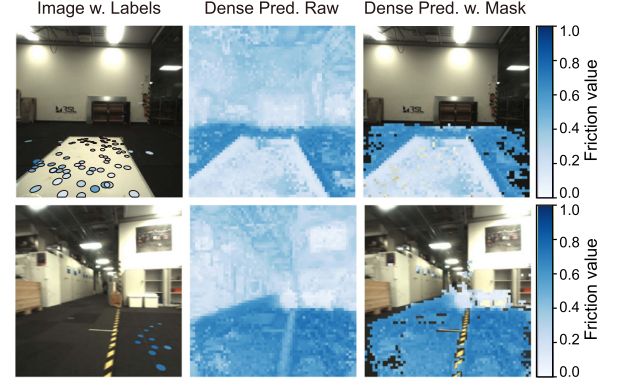


Fig. 11. Dense friction prediction frames. We also output the mean value of friction prediction in the whiteboard/ground areas (Row 1: mask acc. 0.84, Whiteboard pred mean 0.22, Ground pred mean 0.60; Row 2: mask acc. 0.80, Ground pred mean 0.62).

TABLE II  
DENSE PREDICTION ERROR ACROSS THE FULL DATASET

Scene	Fric. pred. error	Conf. mask acc. (IOU)
WB	$0.03 \pm 0.05$	$0.82 \pm 0.05$
GROUND	$0.00 \pm 0.01$	

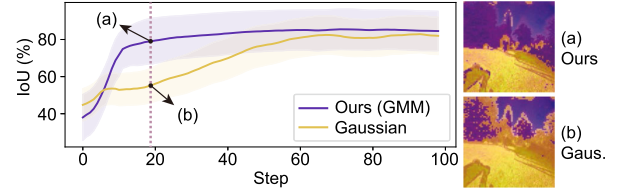


Fig. 12. Confidence mask accuracy development as the increase of training step. Two insets on the right show reconstruction errors visualized in the same image as in Fig. 4 on the 19th step.

the experiment, we aim to evaluate the terrain property prediction and the confidence prediction based on our adaptive anomaly threshold.

An example prediction of our vision pipeline is shown in Fig. 11. To evaluate the confidence mask prediction, we manually label the ground and whiteboard areas within the images as the ground truth mask. The Intersection over Union (IOU) metric for the full dataset is reported in Table II. Our method accurately distinguishes between the floor (including the *WB* and *GROUND* area) and the remaining part of the scene, where most of the predicted physical properties are unreliable, achieving an IOU of 0.82.

From the previous digital twin experiment, we establish a friction range for the *WB* of (0.0,0.2) and (0.3,1.0) for the *GROUND*. The same rule as in Section IV-B applies to the error calculation for each pixel prediction. The mean error for the *WB* area is 0.03 across all images and 0.0 for the *GROUND*, given the large interval of valid friction values.

In addition, we ablated the choice of our proposed GMM method, compared to the original method [18]. We deployed our physical terrain property pipeline on an outdoor dataset, consisting of 82 images, and report the anomaly detection performance. The result in Fig. 12, with two example images, shows that our method can achieve a higher mask accuracy more rapidly than our previous method.

More experimental results of our framework in off-road scenarios and environments with changing physical properties can be found in the supplementary video <https://bit.ly/3Xo5AA8>.

## V. CONCLUSIONS AND DISCUSSIONS

We successfully trained our physical decoder to estimate terrain friction and stiffness from the interaction, outperforming the baseline method with MAE of 0.15 and 0.46 separately. In addition, we showed that the identified parameters of the physical decoder in the real world align with the simulation parameters in the digital twin experiment. Lastly, we demonstrated our online vision pipeline effectively predicts masked terrain friction from vision without interaction.

However, further improvements in terms of prediction stability and inaccuracies are needed. For example, when the robot stands, we observed non-stationary friction estimates, or the change in stiffness for a thin 5 cm foam board in the real world could not be correctly predicted. This is likely due to insufficient training data or noisy real-world geometric observations. We expect by carefully incorporating noise during training our model can be further improved and robustified. Besides, we can enhance the digital-twin-based method to support extensive quantitative analysis of more complex real-world scenarios, beyond a flat and rigid surface.

Our work opens up the possibility of transferring physical-terrain-parameter-aware locomotion and navigation policies, trained in simulation, to the real world, utilizing both visual prediction and proprioception. In the future, we aim to address the limitations and train such policies to make legged robots more capable in the real world.

## REFERENCES

- [1] J. Lee et al., "Learning quadrupedal locomotion over challenging terrain," *Sci. Robot.*, vol. 5, no. 47, 2020, Art. no. ebac5986.
- [2] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "RMA: Rapid motor adaptation for legged robots," in *Proc. Robot.: Sci. Syst. XVII*, D. A. Shell, M. Toussaint, and M. A. Hsieh, Eds., Jul. 2021.
- [3] T. Miki et al., "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Sci. Robot.*, vol. 7, no. 62, 2022, Art. no. eabk2822.
- [4] C. Zhang et al., "Learning agile locomotion on risky terrains," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023.
- [5] M. Tranzatto et al., "Cerberus in the darpa subterranean challenge," *Sci. Robot.*, vol. 7, no. 66, 2022, Art. no. eabp9742.
- [6] T. Miki et al., "Learning to walk in confined spaces using 3D representation," in *IEEE Int. Conf. Robot. Automat.*, 2024, pp. 8649–8656, doi: [10.1109/ICRA57147.2024.10610271](https://doi.org/10.1109/ICRA57147.2024.10610271).
- [7] P. Arm et al., "Scientific exploration of challenging planetary analog environments with a team of legged robots," *Sci. Robot.*, vol. 8, no. 80, 2023, Art. no. eade9548.
- [8] L. Ding et al., "A 2-year locomotive exploration and scientific investigation of the lunar farside by the Yutu-2 rover," *Sci. Robot.*, vol. 7, no. 62, 2022, Art. no. eabj6660.
- [9] L. Wellhausen, R. Ranftl, and M. Hutter, "Safe robot navigation via multi-modal anomaly detection," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1326–1333, Apr. 2020.
- [10] N. Rudin et al., "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Proc. 5th Conf. Robot Learn.*, 2022, vol. 164, pp. 91–100.
- [11] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 895–902, Apr. 2018.
- [12] C. Zhang et al., "Resilient legged local navigation: Learning to traverse with compromised perception end-to-end," in *Proc. 41st IEEE Conf. Robot. Automat.*, 2024, pp. 34–41.
- [13] J. Wu, G. Xin, C. Qi, and Y. Xue, "Learning robust and agile legged locomotion using adversarial motion priors," *IEEE Robot. Automat. Lett.*, vol. 8, no. 8, pp. 4975–4982, Aug. 2023.
- [14] H. Shi et al., "Terrain-aware quadrupedal locomotion via reinforcement learning," 2023, *arXiv:2310.04675*.
- [15] F. Jenelten, J. He, F. Farshidian, and M. Hutter, "DTC: Deep tracking control," *Sci. Robot.*, vol. 9, no. 86, 2024, Art. no. eadh5401.
- [16] P. Xu et al., "Learning physical characteristics like animals for legged robots," *Nat. Sci. Rev.*, vol. 10, no. 5, 2023, Art. no. nwad045.
- [17] S. Choi et al., "Learning quadrupedal locomotion on deformable terrain," *Sci. Robot.*, vol. 8, no. 74, 2023, Art. no. eade2256.
- [18] J. Frey et al., "Fast traversability estimation for wild visual navigation," in *Proc. Robot.: Sci. Syst.*, Jul. 2023.
- [19] M. Mattamala et al., "Wild visual navigation: Fast traversability learning via pre-trained models and online self-supervision," *Auton. Robots*, 2024, *arXiv:2404.07110*.
- [20] J. Seo, S. Sim, and I. Shim, "Learning off-road terrain traversability with self-supervisions only," *IEEE Robot. Automat. Lett.*, vol. 8, no. 8, pp. 4617–4624, Aug. 2023.
- [21] M. G. Castro et al., "How does it feel? Self-supervised costmap learning for off-road vehicle traversability," in *2023 IEEE Int. Conf. Robot. Automat.*, 2022, pp. 931–938.
- [22] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, "Where should I walk? Predicting terrain properties from images via self-supervised learning," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1509–1516, Apr. 2019.
- [23] J. Zürn, W. Burgard, and A. Valada, "Self-supervised visual terrain classification from unsupervised acoustic feature learning," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 466–481, Apr. 2021.
- [24] Z. Yu, S. Perera, H. Hauser, P. R. N. Childs, and T. Nanayakkara, "A tapered whisker-based physical reservoir computing system for mobile robot terrain identification in unstructured environments," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 3608–3615, Apr. 2022.
- [25] B. Gabriel et al., "Learning to see physical properties with active sensing motor policies," in *Proc. 7th Conf. Robot Learn.*, (Proc. Mach. Learn. Res.), J. Tan, M. Toussaint, and K. Darvish, Eds., vol. 229, Nov. 6–9, 2023, pp. 2537–2548. [Online]. Available: <https://proceedings.mlr.press/v229/margolis23a.html>
- [26] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 9630–9640.
- [27] M. Oquab et al., "DINOv2: Learning robust visual features without supervision," *Trans. Mach. Learn. Res.*, 2024. [Online]. Available: <https://openreview.net/forum?id=a68SUt6zFt>
- [28] A. Li, J. Frey, J. Lee, C. Cadena, and M. Hutter, "Seeing through the grass: Semantic pointcloud filter for support surface learning," *IEEE Robot. Automat. Lett.*, vol. 8, no. 11, pp. 7687–7694, Nov. 2023.
- [29] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," in *Proc. Robot.: Sci. Syst.*, Cambridge, Massachusetts, Jul. 2017, doi: [10.15607/RSS.2017.XIII.064](https://doi.org/10.15607/RSS.2017.XIII.064).
- [30] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
- [31] D. Reynolds, *Gaussian Mixture Models*. Boston, MA, USA: Springer, 2009, pp. 659–663.
- [32] P. D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, San Diego, CA, USA, 2015, *arXiv:1412.6980*.
- [33] M. Bloesch et al., "State estimation for legged robots: Consistent fusion of leg kinematics and IMU," in *Proc. Robot.: Sci. Syst.*, Sydney, Australia, Jul. 2012, doi: [10.15607/RSS.2012.VIII.003](https://doi.org/10.15607/RSS.2012.VIII.003).
- [34] G. Haddeler et al., "Real-time digital double framework to predict collapsible terrains for legged robots," in *2022 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 10387–10394.