See-Touch-Predict: Active Exploration and Online Perception of Terrain Physics with Legged Robots

Huangxuan Lin[1], Haoyang Li[1], Yue Gao[2†]

***Abstract*—Assessing physical properties of the environment with vision helps humans to respond appropriately before entering risky areas. However, equipping robots with such perceptual ability is challenging due to the lack of labeled data. To overcome this challenge, we present the Active Exploration and Online Perception (AEOP) framework, enabling legged robots to actively estimate and predict physical information of unknown terrains in real-time. The framework contains a learning-based physical sensing policy that controls the robot to actively estimate terrain physics such as friction coefficient and an incremental terrain classifier which performs temporal and spatial consistent terrain segmentation based on color images. A mapping module conjugates physical properties and visual categories of different terrains, building a point cloud map labeled with physical properties at run-time. Extensive real-world experiments were conducted where the proposed framework correctly distinguished a wide range of terrain from slippery ($\mu = 0.17$) to rough ($\mu = 1.03$) and accurately predicted the friction coefficient of encountered terrains based on vision, providing efficient perception of terrain physics for legged robots.**

***Index Terms*—Legged Robots; Deep Learning for Visual Perception**

I. INTRODUCTION

In recent years, legged robots have shown remarkable ability to traverse challenging terrains. State-of-the-art legged robot control methods utilize proprioception and depth sensing to perceive geometry information and obstacles, enabling robots to perform robust locomotion over a wide variety of terrains [1]–[7], flexibly avoid obstacles [8], [9], and even parkour in extreme environments [10], [11]. The key to robust legged locomotion is the cognition and adaptation of diverse physical conditions in the environment. Earlier works proposed domain randomization as a way to improve the generalization of policy by training it under a wide range of physical conditions and observation noise [12]–[14]. Follow-up works extract implicit features from sequential historical observations as policy inputs to make more effective distinctions between different physical environments [3]–[5], [15].

Manuscript received: October, 14, 2024; Revised January, 1, 2025; Accepted January, 26, 2024.

This paper was recommended for publication by Editor Abderrahmane Kheddar upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the National Natural Science Foundation of China (Grant No. 62373242 and No.92248303),Shanghai Municipal Science and Technology Major Project (Grant No. 2021SHZDZX0102).

[1]Huangxuan Lin and Haoyang Li are with Department of Automation, Shanghai Jiao Tong University, Shanghai, P.R. China. {b3mylq, Lhy_0415}@sjtu.edu.cn. [2]Yue Gao is with MoE Key Lab of Artificial Intelligence and AI Institute, Shanghai Jiao Tong University, P.R. China, yuegao@sjtu.edu.cn. †Yue Gao is the corresponding author.

Digital Object Identifier (DOI): see top of this page.

Implicit features that highly coupled to the control policy are difficult to reuse for new controllers and tasks. To obtain multi-purpose representation, it is intuitive to explore explicit parameters that describe the terrain's intrinsic nature, such as friction and stiffness. Many previous studies have demonstrated the ability of legged robots to classify terrains [16], [17] or estimate physical parameters of the ground [4], [18], [19] from mechanical sensor data through interaction with the terrain. However, the perception through mechanical sensors requires a period of movement in unknown environments, such passive and delayed sensing poses a significant risk under drastically changing dynamic conditions.

Assessing physical properties of the environment with vision helps humans to respond appropriately before entering risky areas. Yet training robots with such perceptual ability using supervised learning is not feasible at this stage as image datasets labeled with accurate physical descriptions are highly challenging to construct. To utilize color images for motion control, previous works mainly adopt self-supervised methods to label color images with terrain properties collected in real world [18], [20]–[22]. A typical implementation is to project the estimated metrics from the footholds to the image space as labels for the corresponding pixels, thus training the visual network to predict these metrics (e.g. traversability or other proxy scores) [20]–[22]. [18] presents an active sensing motor policy that estimates terrain's physical parameters, collecting paired physics and vision data to train a pixel-wise estimator for predicting terrain physics from color images.

However, these self-supervised methods require re-training from online collected data, significantly affecting the efficiency of terrain assessment in unknown environments. Additionally, legged robots interact with only a small part of the ground, causing the sparsity of data obtained from foot-ground contacts which leads to slow training and high prediction noise.

In this work, we introduce a framework for legged robots to actively estimate and predict physical information of unknown terrains in real-time. The framework includes a physical sensing policy, an incremental terrain classifier, and a mapping module. The physical sensing policy controls the robot to actively interact with the terrain and estimate physical parameters. The incremental terrain classifier performs temporal and spatial consistent terrain segmentation based on color images. The mapping module conjugates texture category and physical parameters of terrains to form a point cloud map labeled with physical properties.

The key contributions of our work can be listed as follows:

- We proposed the Active Exploration and Online Perception (AEOP) framework to achieve real-time understand-

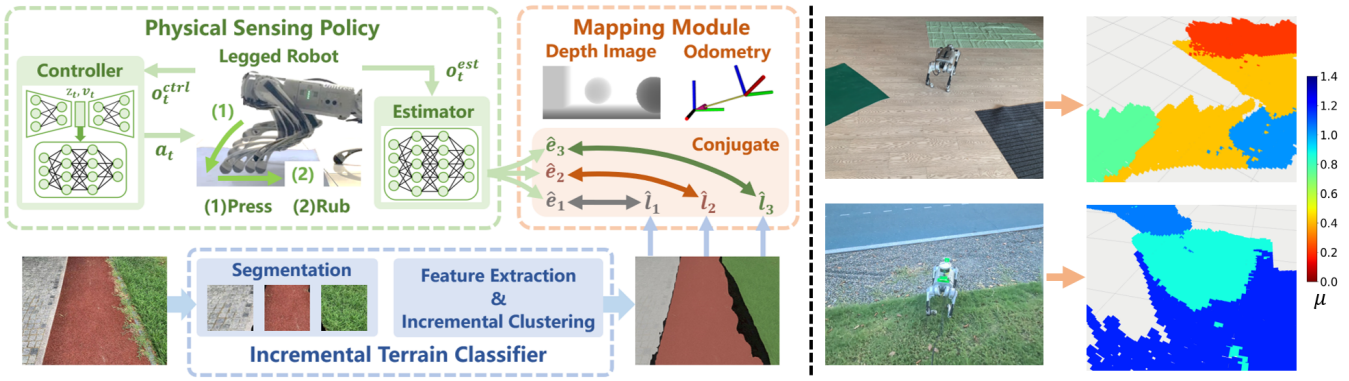

Fig. 1: **Active Exploration and Online Perception Framework.** *Left*: Overview of the proposed framework. The mapping module conjugates physical parameters from physical sensing policy (Sec. III) and terrain segmentation image from incremental terrain classifier (Sec. IV) to build a point cloud map labeled with physical properties. *Right*: Mapping results in real-world experiments, where μ represents the friction coefficient of the ground.

ing of physical environment for legged robots. To the best of our knowledge, this is the first time that legged robots can predict physical parameters of the terrain without additional training during deployment.

- A learning-based physical sensing policy that enables legged robots to perform active probing motions and estimate friction coefficient of the ground.
- A incremental terrain classifier that segments terrain textures from color images in unknown environment, without relying on pre-specified categories.
- We deploy the AEOP framework on the Unitree Go1 quadruped robot to perform accurate and efficient physical perception in both indoor and outdoor environments, validating the effectiveness of the above method.

II. System Overview

The goal of this work is to design a perception system for legged robots to estimate and predict terrains' physical properties. We aim for a system that relies only on standard proprioception sensors, requires no terrain type annotation, and doesn't need real-world data for retraining.

The framework of our method is shown in Fig. 1, which consists of a physical sensing policy, an incremental terrain classifier, and a mapping module. The physical sensing policy controls the robot to fully interact with the terrain using deep reinforcement learning (DRL) and evaluates physical properties of the contact location (further introduced in Sec. III). The incremental terrain classifier uses a pre-trained model to segment different terrains from RGB images. A contrastive learning network extracts texture features for each terrain to give temporal and spatial consistent labels through incremental clustering (further introduced in Sec. IV).

Receiving the terrain segmentation image from incremental terrain classifier, the mapping module uses an aligned depth image, camera intrinsics and a Lidar odometry to build a point cloud map labeled with terrain categories. When the robot stands on a terrain with unknown physical properties, the mapping module prompts the operator to employ physical sensing policy and estimate physical parameters of the terrain. The estimated parameters are assigned to the texture class of that terrain to predict the physical properties of all similar terrains.

III. Physical Sensing Policy

This section introduces the physical sensing policy which aims to perform accurate ground physical parameter estimation. The policy commands the robot to perform probing motions(Sec. III-A) using a DRL based end-effector tracking controller(Sec. III-B), providing effective proprioception for physical parameter estimation(Sec. III-C).

A. Probing Motion Design

As mentioned in previous work [18], the robot's behaviour significantly affects its perception of the physical environment. For example, when the robot walks across rough terrains without slipping, it is difficult to make an accurate estimation of the surface friction coefficient. Inspired by the tactile motions proposed in [19], we designed a three phase probing motion to guide the robot to make a digging-like movement, enabling full interaction with the terrain in both the normal and tangential directions:

1) Pressing in Normal Direction: Robot's foot starts pressing downward from the overhanging position at the front of the shoulder link (Fig. 2-(a)). Considering the undulation of the terrain, the robot's foot keep moving downward until the magnitude of its contact force with the ground reaches a certain threshold value.

2) Rubbing in Tangential Direction: The robot's foot maintains specified contact force with the ground while rubbing back tangentially along the ground (Fig. 2-(b)).

3) Reset: Robot's foot smoothly returns to the position before normal pressing stage.

The probing motions are realised by commanding the robot to reach a sequence of the target position command $\mathcal{T} = \{c_0, ..., c_{n-1}, c_n\}$ using its forefoot, where $c_t \in \mathbb{R}^3$ represents the coordinates of the target position in robot's body frame. The contact force F between robot's foot and the ground is estimated using rigid body dynamics:

$$\hat{F} = (J^T)^{-1}\tau \tag{1}$$

Fig. 2: Demonstration of the probing motions.

where J is the Jacobian matrix of the robot's foot relative to the robot's body frame. In the first two phases the z-component of c_n is updated using a simple proportional controller to press down the foot during pressing phase and maintain specified contact force during the rubbing phase:

$$z_{n+1} = z_n + clip[k_p(\hat{F}_z - F_t), -0.01m, 0.01m] \quad (2)$$

where $\hat{F}_z$ is the z-component of the estimated contact force, target force $F_t = 15N$, proportional coefficient $k_p = 0.001$.

B. End-effector Tracking Controller

The end-effector tracking controller π_c enables legged robot to reach target position c_t with one of its end-effectors (i.e., the robot's feet) while maintaining stable with other legs. The controller π_c is trained using Proximal Policy Optimization (PPO).

1) Policy Input: Input of the control policy o_t^{ctrl} includes past 20-timesteps history of the proprioception states s_t^{ctrl}, where s_t^{ctrl} is a 58-D vector consists of: body angular velocity $\omega_t \in \mathbb{R}^3$, gravity vector in body frame $g_t \in \mathbb{R}^3$, joint position $q_t \in \mathbb{R}^{12}$, joint velocity $\dot{q}_t \in \mathbb{R}^{12}$, last action $a_{t-1} \in \mathbb{R}^{12}$, feet positions in body frame $x_t \in \mathbb{R}^{12}$, distance between robot's left forefoot and target position $d_t \in R$ and target position command $c_t \in \mathbb{R}^3$. The command is sampled once per second.

2) Action Space: Output of the control policy $a_t \in \mathbb{R}^{12}$ is the desired joint positions, followed by a PD controller with $k_p = 20$ and $k_d = 0.5$ to obtain joint torques τ_t.

3) Architecture: We adopt the same asymmetric actor-critic architecture used in [3]. The actor consists of a context-aided estimator network implemented using β-variational auto-encoder (β-VAE) and an MLP-based policy network. The critic receives proprioception states s_t^{ctrl} and height map scan of the robot's surroundings h_t to estimate state value. The hyper-parameter β is set to 4.

4) Rewards: The reward functions consist of three main components. Task rewards encourage the end-effector to quickly approach and remain at the target position. Inspired by previous works in locomotion task [3], [5], [23], we apply auxiliary rewards to constrain robot's motion and smoothing rewards to ensure smooth and efficient robot movement. Weights of the task rewards are carefully tuned to reach a balance between position tracking and posture stability. Details of the reward functions are given in Table I.

C. Physical Parameter Estimator

An MLP-based estimator network is trained to estimate physical parameters of the terrain using robot proprioception when the robot is performing probing motions defined by the command sequence $\mathcal{T}$.

TABLE I: Reward List of End-effector Tracking.

Term	Reward	Equation	Weight
Task	Target position	e^{-4d_t}	3.
	Target velocity	$d_{t-1} - d_t$	25.
Auxiliary	Lin. velocity(z)	v_z^2	-2.0
	Ang. velocity(xy)	ω_{xy}^2	-0.05
	Orientation	$\|g\|^2$	-0.2
	Body height	$(h^{des} - h)^2$	-5.0
	Collision	$n_{collision}$	-1.0
Smoothing	Action smoothing	$(a_t - a_{t-1})^2$	-0.05
	Action smoothing2	$(a_t - 2a_{t-1} + a_{t-2})^2$	-0.01
	Joint accelerations	$\ddot{\theta}^2$	-2.5e-7
	Joint power	$\|\tau\|\|\dot{\theta}\|$	-2.5e-5

TABLE II: Environment Settings of Physical Sensing Policy

Parameters	Range	Parameters	Range
Friction Coefficient	[0.1, 1.5]	target position x	[0.06m, 0.36m]
Restitution	[0, 0.5]	target position y	[0.12m, 0.24m]
Terrain Undulation	[0m, 0.1m]	target position z	[-0.4m, -0.12m]

The estimator's input o_t^{est} includes past 20-timesteps history of estimation states s_t^{est} defined as follows:

$$s_t^{est} = \begin{bmatrix} \omega_t & g_t & q_t & \dot{q}_t & \tau_t & x_t & n_t \end{bmatrix} \quad (3)$$

where n_t is a boolean quantity representing whether robot's tracking foot touches the ground.

Considering that observations obtained when robot's foot is in the air cannot be effectively used for parameter estimation, only when a minimum of 80% of the n_t in an input sequence o_t^{est} is true that o_t^{est} is considered a valid observation and used for estimator training.

The estimator outputs physical parameter e_t of the contact location. In our experiments, e_t is the ground friction coefficient u_t. To facilitate the convergence of training, the ground truth of u_t is discretized into 20 bins according to the sampling range. The estimator outputs a 20-D vector which is the logits of u_t and is trained with cross-entropy loss.

D. Implementation Details

Both the end-effector tracking controller and the physical parameter estimator are trained with 2048 agents in parallel for 5000 episodes in Isaac Gym simulator [24]. The training environment consists of 50*50 squares of size 3m*3m stitched together, each with randomly selected friction coefficient, restitution coefficient and terrain. Every 20s each robot is regenerated to a new randomly selected location to fully experience the terrain with various physical parameters. Details of the environment settings are listed in Table II.

IV. Incremental Terrain Classifier

This section describes how to incrementally segment RGB image of different terrains with temporal and spatial consistent labels. The classifier consists of a pre-trained segmentation model (Sec. IV-A), a texture feature extraction network (Sec. IV-B), and an incremental cluster (Sec. IV-C). The classifier inputs an RGB image $I \in \mathbb{R}^{w \times h \times 3}$, outputs a segmented image $H \in \mathbb{R}^{w \times h}$ where the value of each pixel represents the terrain type to which the pixel belongs.

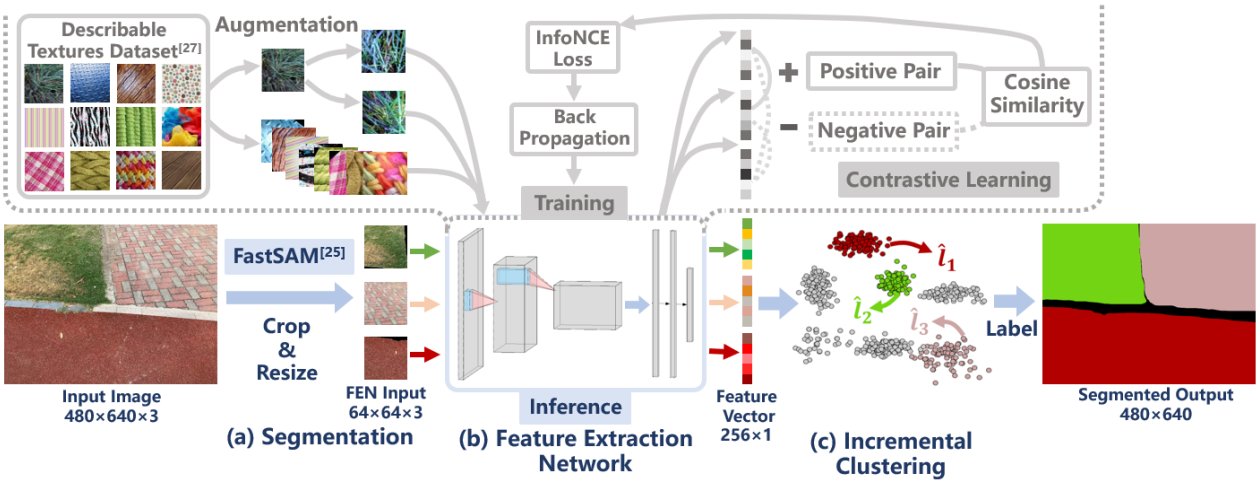

Fig. 3: **Overview of the Incremental Terrain Classifier.** Blue arrows indicate the deployment process of the classifier: (a). FastSAM performs fine-grained segmentation for different terrains. (b): The feature extraction network trained by contrastive learning extracts feature vectors for each terrain. The training framework is indicated by gray arrows. (c): feature vectors from terrain textures is used to generate pseudo-label through incremental clustering.

A. Segmentation

Given the RGB image I of the ground mixed by various terrains, the segmentation model outputs pixel-wise mask $T \in \mathbb{R}^{w \times h \times k}$ for k terrain categories. FastSAM [25] is chosen as the segmentation model, a lightweight CNN model built to perform real-time segment-anything task. All-instance mode segmentation is used to avoid human intervention. The fineness of segmentation is mainly influenced by the hyperparameters IoU threshold and confidence threshold. Segmentation results may appear to be erroneous without oversight, including segmenting different terrains into the same category or segmenting the same type of terrain into different categories. The former error should be avoided while the latter is tolerable as labels will be corrected by the clustering module. Therefore, the IoU threshold is set to 0.9 and the confidence threshold is set to 0.5, encouraging finer segmentation.

B. Feature Extraction Network

The labels given by the segmentation model are not temporally and spatially consistent, where pixels belonging to the same category in images captured from different poses and times may not have the same label. To retain physical properties assigned to the label of terrain from past observations, it is necessary to determine whether the segmented parts of input images belong to the same type of terrain. Following this intuition, we propose the Feature Extraction Network (FEN) to extract feature vectors from various terrain textures and performing incremental clustering to obtain consistent pseudo-labels for each terrain.

The FEN aims to obtain distinguishable feature vectors for different terrain textures to enhance clustering accuracy. The network leverages contrastive learning to minimize the similarity between different textures and maximize it between similar textures. The FEN uses ResNet-18 [26] as backbone encoder $f(\cdot)$, concatenated by a multi-layer perceptron (MLP) projection head $g(\cdot)$ containing one hidden layer. The FEN is trained on Describable Textures Dataset [27] which contains 5,640 pictures with different textures. Implementation details of the training are listed below:

1) Data Augmentation: Typical augmentation strategies used in contrastive learning include random crop and resize, random color distort, random Gaussian blur, etc. However, robots are confronted with environments containing a wider range of visual transformations, primarily including changes in observation poses and various types of local highlights caused by different light sources. Therefore, we incorporate random affine transformations and random local highlighting to improve the robustness of FEN. We implement affine transformation by applying random rotation, scale and shear to the image while superimposing two-dimensional Gaussian distribution offset to the color channels for local highlighting.

2) Loss Function: For a training batch $\{x_k\}_{k=1}^{N}$, each raw image x_k generates two augmented images $\tilde{x}_{2k-1}$ and $\tilde{x}_{2k}$ through data augmentation, which serves as positive pairs to each other. Each pair of positive samples treats other 2(N-1) augmented images as their negative pair. The FEN inputs augmented image $\tilde{x}$ and outputs feature vector z where $z_i = g(f(\tilde{x}_i))$. The similarity between two feature vectors is defined using cosine similarity:

$$\mathrm{sim}(z_i, z_j) = \frac{z_i \cdot z_j}{\|z_i\| \|z_j\|} \tag{4}$$

Similar to the approach of SimCLR [28], the loss function is defined using infoNCE loss. For a positive pair (i, j) the loss $\mathcal{L}(i, j)$ is calculated by:

$$\mathcal{L}(i,j) = -\log \frac{\exp(\mathrm{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(z_i, z_k)/\tau)} \tag{5}$$

where $\mathbf{1}_{[k \neq i]} \in \{0, 1\}$ denotes an indicator function whose value is set as 1 if $k \neq i$ and τ represents the temperature parameter. The final loss is computed across all positive pairs of a training batch, both $\mathcal{L}(i, j)$ and $\mathcal{L}(j, i)$, encouraging the

similarity between positive pairs to approach 1, while the similarity between negative pairs tends towards 0:

$$\mathcal{L}_{\text{batch}} = \frac{1}{2N}\sum_{k=1}^{N}[\mathcal{L}(2k-1,2k)+\mathcal{L}(2k,2k-1)] \quad (6)$$

The input size of FEN is $64 \times 64 \times 3$. For each segmented portion of raw image I, we utilize a customized cropping algorithm that iteratively moves the boundary towards the center of the segmented pixels to obtain a rectangular sub-image with sufficient segmented pixels. Each sub-image generates 4 input samples to FEN by performing the same random crop and resize operation in data augmentation.

C. Incremental Clustering

After extracting feature vectors from terrain textures, the IDBSCAN algorithm [29] is used to cluster feature vectors to obtain pseudo-labels for each terrain. IDBSCAN is a density-based incremental clustering algorithm. It can determine the number of clusters based on the distribution of data, making it suitable for environments with unknown terrain types.

Typically, IDBSCAN employs Euclidean distance as the distance criterion, which becomes less discriminative in high-dimensional space and thus significantly impacts clustering performance. Meanwhile, the algorithm performs poorly when there are significant disparities in density between clusters. Considering the property of FEN, the distance criterion between two feature vectors z_i and z_j is defined using cosine similarity:

$$\mathcal{D}(i,j) = 1 - \text{sim}(z_i, z_j) \quad (7)$$

This criterion ensures that the distances between feature vectors of the same class fall within a well-defined range and remain effective in high-dimensional spaces.

V. Experiments

We conducted extensive experiments to verify the effectiveness of the proposed framework and each of its modules.

All experiments are conducted on Unitree Go1 quadrupedal robot. A RealSense D435 camera is used to acquire aligned RGB-D images while the odometry is obtained using a Velodyne VLP-16 Lidar with LeGO-LOAM [30], a lightweight SLAM Algorithm. All modules are trained on a desktop with NVIDIA RTX 3090 GPU and are deployed on a laptop with NVIDIA RTX 3070 GPU and Intel i7-11800H CPU. The physical sensing policy and motion control run on a frequency of 50 Hz and the terrain classifier runs on 2 Hz.

A. Physical Parameter Estimation

We designed real-world experiments to validate the effectiveness of the physical sensing policy.

1) Experiment Setup: The experiment aims to evaluate the accuracy of the strategy in estimating physical parameters (taking the friction coefficient as an example). The selected test materials covers a wide range of friction coefficients commonly encountered in everyday life, from slippery (μ = 0.17) to rough (μ = 1.03). We manually measured the friction coefficients of these materials as the ground truth, using the equipment setup shown in Fig. 4. In the experiment, some materials are laid out on uneven terrain to test the robustness of the policy in outdoor environments.

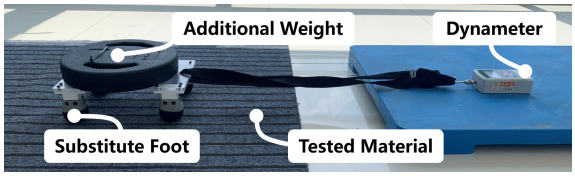

Fig. 4: Equipment setup used to measure the ground truth of friction coefficient

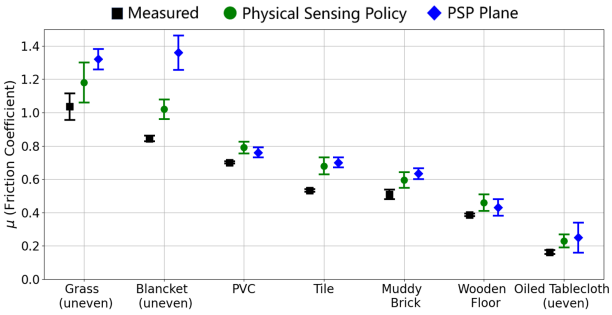

Fig. 5: **Real-world friction estimation performance.** The proposed Physical Sensing Policy obtained more stable and accurate estimation (green) to the measured ground truth (black) compared to that trained only on flat terrain (PSP Plane, blue).

2) Results: The experimental results are shown in Fig. 5. It can be seen that the proposed physical sensing policy is effective in distinguishing materials with different friction coefficients. Compared to the physical sensing policy trained only on flat terrain (PSP Plane), the training on diverse uneven terrain significantly improves the robustness of the policy to the undulations of the real-world terrain. Though the estimator is trained only on rigid surface, it is capable of assessing some terrains that exhibit non-rigid characteristics, provided that the frictional properties of the terrain remain approximately constant during the sliding. The experimental results confirmed this statement by accurate estimation on non-rigid surfaces like grass and foam mat, contaminated surfaces like muddy brick and oiled tablecloth, and granular surfaces like gravel (shown in the supplementary video).

We also conducted evaluation of physical sensing policy in simulation under the similar environment setting of Sec. III-D. The measured error distributions and mean absolute errors are shown in Fig. 7 using green box plots and orange bar charts respectively, demonstrating high accuracy and strong robustness of the proposed estimator in simulation.

Compared to the results of the simulation experiments, the results in real-world experiments exist systematic bias in high-friction region, indicating the existence of sim-to-real gap in current method with respect to robot dynamics and terrain modelling. We suspect that real-world motors are subject to greater damping than the PD model in simulation, which requires higher torque output to perform probing motions and make the robot overestimates the friction coefficient. Nevertheless, the estimated parameters correctly reflect the relative relationship between material friction properties from slippery to rough, and are particularly accurate in assessing slippery materials. Thus, they can provide reliable and valuable

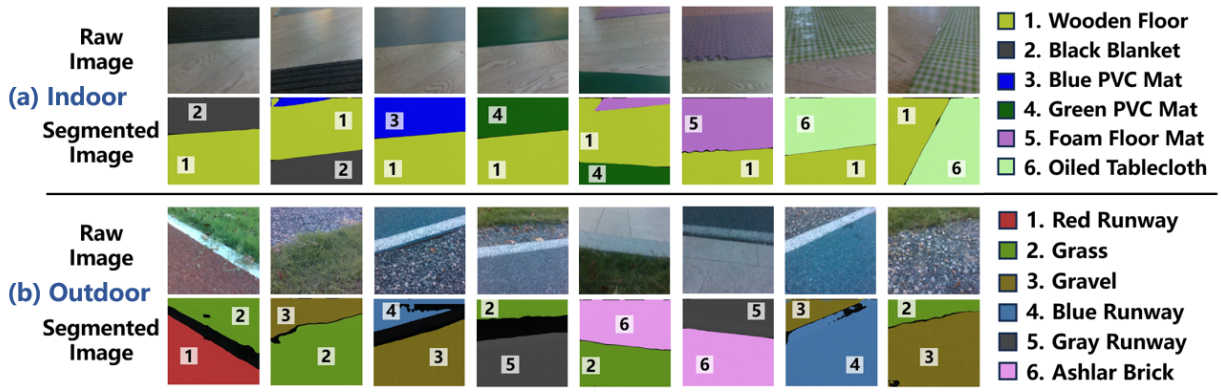

Fig. 6: Segmentation results of incremental terrain classifier.

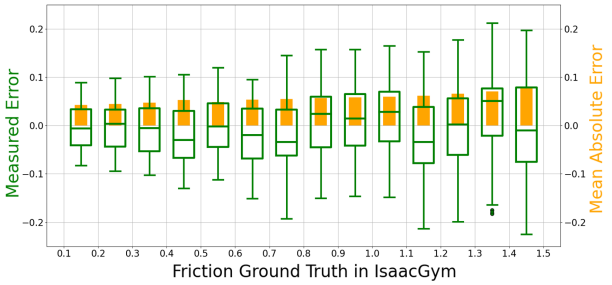

Fig. 7: Friction estimation performance in IssacGym.

prior observations for control strategies.

B. Terrain Clustering and Segmentation

1) Clustering Task: The accuracy of terrain clustering is evaluated using 36 sets of RGB images collected indoors and outdoors. The dataset contains 21 commonly found terrain classes, including but not limited to the terrains illustrated in Fig. 9. For each set, the robot circled a location to capture images from various poses. Some terrain classes were collected at different locations and times to test the network's robustness to lighting conditions.

2) Baseline: To reveal the effect of different training designs on terrain clustering performance, we conduct ablation experiments on the following baselines:

- **Retrained Network:** FEN trained on the evaluation dataset using the same setup as the proposed approach.
- **FEN w/o Neg. pairs:** Train FEN without using negative pairs, following the similar approach to SimSiam [31].
- **FEN w/o customized Aug.:** Train FEN without affine transformation and random highlighting.

3) Metrics: The following metrics are used to evaluate the performance of clustering:

- **Clustering accuracy (Acc.):** The ratio of accurate clustering to all terrain types. A clustering is defined as an accurate clustering if all sets of a same terrain type are clustered into the same category.
- **Mild error rate (Mer.):** A clustering is defined as a mild error if pictures of the same terrain, taken in different conditions, are clustered into different categories. Such an error affects perception efficiency but does not cause the robot to misjudge the physical properties.

TABLE III: Clustering Performance

Method	Online	Acc. ↑	Mer. ↓	Ser. ↓
Retrained Network	✗	19/21	2/21	0/21
FEN w/o Neg. Pairs	✓	7/21	8/21	6/21
FEN w/o Customized Aug.	✓	14/21	7/21	0/21
FEN	✓	**18/21**	**3/21**	**0/21**

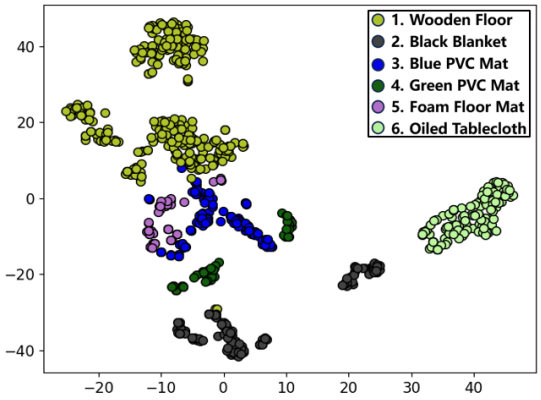

Fig. 8: t-SNE plot of the FEN's output z from terrains in Fig. 6-(a), different terrains are clearly distinguished

- **Severe error rate (Ser.):** A severe error refers to the situation where different terrains are clustered into the same category. Such an error can cause misjudgements to the physical environment and affect the safety of motion control.

4) Clustering Performance: The experiment results are summarized in Table III, showing that the use of negative pairs and customized augmentation design significantly improves the clustering performance, bringing the FEN to a level close to that of networks trained on evaluation dataset.

Although self-supervised learning that does not rely on negative pairs has been extensively studied, negative pairs are necessary for our task as density-based clustering inherently requires that features of the same textures are close together, while those of different textures are far apart. The results also validate the effectiveness of affine transformation and highlighting in data augmentation, as the network trained with original data augmentation could not effectively aggregate similar textures under varying lighting conditions or anisotropic

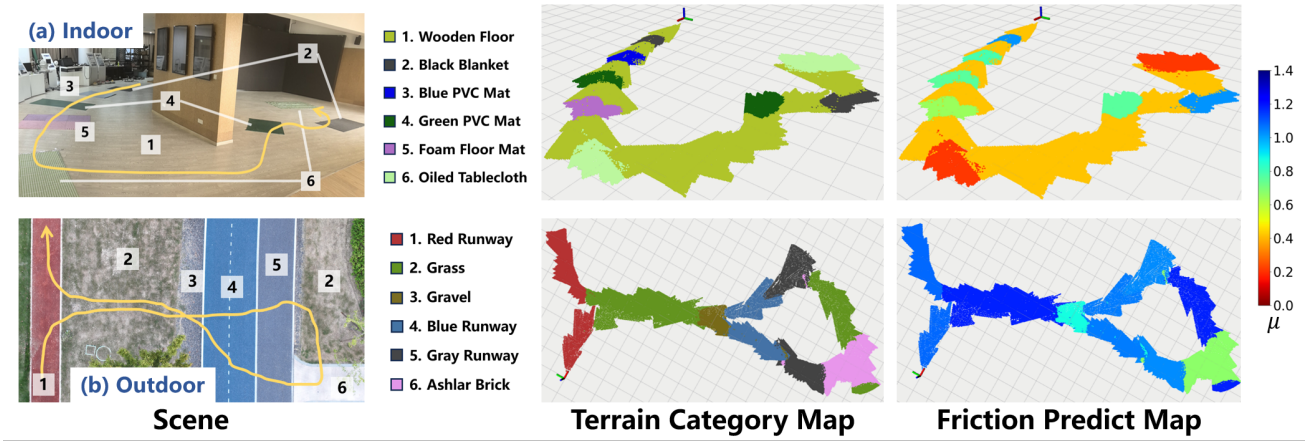

Fig. 9: **Results of integrate system deployments.** (a): The system is deployed in indoor environments (left) and can accurately distinguish terrain textures (middle) and physical parameters (right). (b): The system is deployed in outdoor environments (left) and can accurately distinguish terrain textures (middle) and physical parameters (right).

textures from different observation angles (such as spliced wood boards).

5) Segmentation Results: Fig. 6 shows the results of incremental terrain classifier at different time steps in indoor and outdoor integrated system experiments (Sec. V-C). It can be seen that different types of terrain are properly segmented under varying observation poses and lighting conditions. We preserved all feature vectors z extracted in the indoor experiments and used t-SNE [32] to categorize these vectors (Fig. 8), demonstrating that the feature vectors obtained from the FEN indeed possess good classifiability.

C. Integrated System Deployments

1) Experiment Setup: The integrated system illustrated in Fig. 1 was deployed in various environments to validate its ability in exploring and predicting different terrains' physical properties. For the indoor experiment, the scenario and route are depicted in Fig. 9 (a), where we constructed the floor with a variety of materials, covering a wide range of friction coefficients. The scenario and route of the outdoor experiment are shown in Fig. 9 (b), which contains more complex visual textures and terrain undulations, posing a challenge to the system's robustness. The robot was operated to move along the designated route while identifying the type of terrain observed via color images. For terrain that had not been encountered, the robot estimates the friction coefficients of the terrain using the physical sensing policy. The estimation results are then used to predict the friction coefficients of similar terrains, establishing a point cloud map annotated with physical information.

2) Results: The mapping results of the indoor and outdoor experiments are shown in Fig. 9. The integrated system accurately distinguishes different types of terrain and can predict the physical properties of them using the established correspondence between terrain categories and friction coefficients. The system runs in real time and requires no further training during the deployment. We built a 2D map where all points are projected to the same height for simplicity.

Fig. 10 selects certain timestamps from the indoor experiment to showcase the process of physical exploration and

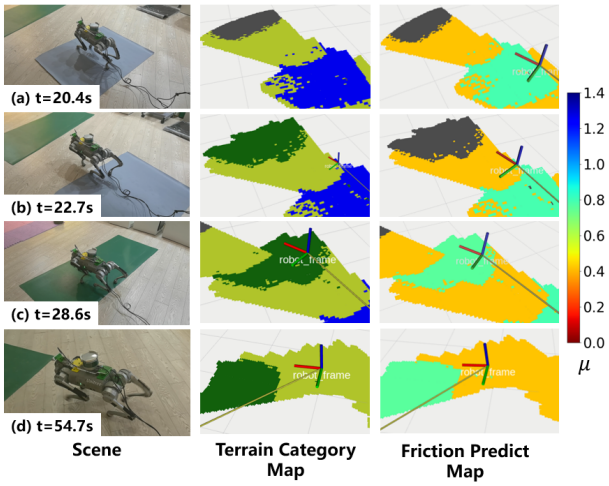

Fig. 10: **Detailed process of physical exploration and perception.** (a) robot encounters a new type of terrain (the green PVC mat), labeled with gray point cloud. (b) the terrain is clustered into a new visual type. (c) the friction coefficient μ of the terrain is obtained. (d) robot predicts the friction coefficient of the terrain.

mapping. When encountering a new type of terrain, where the robot has not collected a sufficient number of visual samples nor conducted physical parameter identification, all textures are classified as "noise", and their physical information is labeled as "unknown" (both shown as gray point clouds in Fig. 10-(a)). When the system has collected enough ground texture samples, a new cluster will be formed for the terrain, which is annotated on the terrain category map (green point clouds in Fig. 10-(b)). At each timestamp the system checks for points within a rectangular region surrounding the estimation foot, if the proportion of points with unknown friction coefficients exceeds 90% of the region and these points are from the same terrain category, the operator will command the robot to execute physical sensing policy and allocate the estimated friction coefficient to the corresponding terrain, generating a point cloud map annotated with physical properties (Fig.

10-(c)). This information can then be used to predict the friction coefficients of similar terrains ahead (Fig. 10-(d)). More experimental details can be found in the supplementary video.

VI. Conclusions

A. Discussions

This paper presents a perceptual framework that enables legged robots to explore and predict terrain physics. While the proposed framework has been successfully deployed in rich scenarios, further improvements to extend robots' capability in physical sensing and visual cognition are worth studying. Our work demonstrates the abilities of legged robots to explore physical information taking friction coefficient as an example, introduction of richer representations of terrain physics (stiffness, granularity, etc.) will certainly enhance the accuracy and effectiveness of physical sensing. In addition, the incremental terrain classifier uses only texture information to classify terrain, whose accuracy decreases as the scene scale and the variety of terrains increases. Future work will incorporate semantic, temporal, and spatial information to establish a more stable classification.

B. Limitations

In current experiments that already contains most of the terrain textures in urban environment, the incremental terrain classifier is able to avoid severe errors defined in Sec. V-B. However, the avoidance of such errors are not guaranteed due to the inherent limitation of clustering algorithms and we temporarily lack a comprehensive assessment for the capability limitation of terrain classifiers. We have started our work to create a larger scale benchmark based on the proposed approach. Another improvement in progress is to handle failure cases with recovery perception. So far the approach naively assign the estimated property to the corresponding terrain and is unable to recover from potentially errors. An intuitive solution is to add physical property evaluation during the motion control and introducing statistical methods to update the terrain physical parameters.

References

[1] J. Lee and et al., "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[2] T. Miki and et al., "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[3] I. M. A. Nahrendra and et al., "Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5078–5084.

[4] S. Choi and et al., "Learning quadrupedal locomotion on deformable terrain," *Science Robotics*, vol. 8, no. 74, p. eade2256, 2023.

[5] Z. Luo and et al., "Moral: Learning morphologically adaptive locomotion controller for quadrupedal robots on challenging terrains," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4019–4026, 2024.

[6] A. Agarwal, A. Kumar, J. Malik, and D. Pathak, "Legged locomotion in challenging terrains using egocentric vision," in *6th Annual Conference on Robot Learning*, 2022.

[7] R. Yang and et al., "Neural volumetric memory for visual locomotion control," in *Conference on Computer Vision and Pattern Recognition 2023*, 2023.

[8] T. He and et al., "Agile but safe: Learning collision-free high-speed legged locomotion," in *Robotics: Science and Systems (RSS)*, 2024.

[9] R. Yang and et al., "Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers," in *International Conference on Learning Representations*, 2022.

[10] X. Cheng and et al., "Extreme parkour with legged robots," *arXiv preprint arXiv:2309.14341*, 2023.

[11] Z. Zhuang and et al., "Robot parkour learning," in *Conference on Robot Learning (CoRL)*, 2023.

[12] J. Tobin and et al., "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

[13] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3803–3810.

[14] J. Hwangbo and et al., "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[15] A. Kumar and et al., "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[16] M. Bednarek and et al., "Fast haptic terrain classification for legged robots using transformer," in *2021 European Conference on Mobile Robots (ECMR)*, 2021, pp. 1–7.

[17] M. Bednarek, M. R. Nowicki, and K. Walas, "Haptr2: Improved haptic transformer for legged robots' terrain classification," *Robotics and Autonomous Systems*, vol. 158, p. 104236, 2022.

[18] G. B. Margolis and et al., "Learning to see physical properties with active sensing motor policies," *Conference on Robot Learning*, 2023.

[19] L. Ding and et al., "Pressing and rubbing: Physics-informed features facilitate haptic terrain classification for legged robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 5990–5997, 2022.

[20] L. Wellhausen and et al., "Where should i walk? predicting terrain properties from images via self-supervised learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1509–1516, 2019.

[21] M. Guaman Castro and et al., "How does it feel? self-supervised costmap learning for off-road vehicle traversability," IEEE, 2023.

[22] J. Frey and et al., "Fast Traversability Estimation for Wild Visual Navigation," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

[23] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," *Conference on Robot Learning*, 2022.

[24] V. Makoviychuk and et al., "Isaac gym: High performance gpu-based physics simulation for robot learning," 2021.

[25] X. Zhao and et al., "Fast segment anything," 2023.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[27] M. Cimpoi and et al., "Describing textures in the wild," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[28] T. Chen and et al., "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML'20. JMLR.org, 2020.

[29] M. Ester and et al., "Incremental clustering for mining in a data warehousing environment," in *Proceedings of the 24rd International Conference on Very Large Data Bases*, ser. VLDB '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, p. 323–333.

[30] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4758–4765.

[31] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15 750–15 758.

[32] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.