Learning Physically Grounded Robot Vision with Active Sensing Motor Policies

Anonymous Author(s)
Affiliation
Address
email

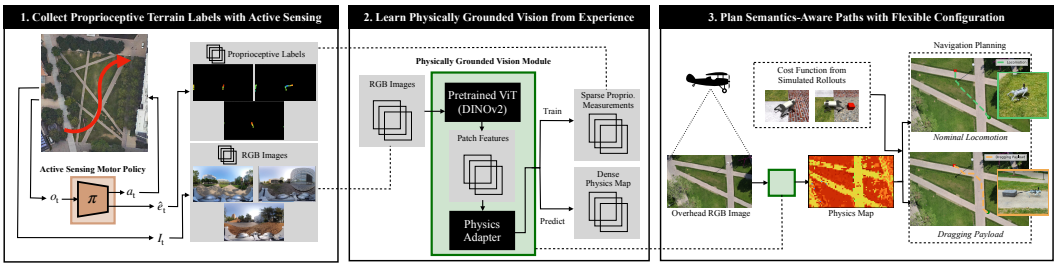


Figure 1: **Learning to see how terrains feel.** We propose (1) learning an optimized gait for collecting informative proprioceptive terrain labels that are (2) used to supervise training for a vision module, which can (3) be used for navigation planning with new tasks and image sources.

Abstract: We present a novel approach to grounding visual perception in the physics of robot locomotion. Our method uses a small amount of real-world interaction data to train a visual perception module, where the labels are derived from proprioceptive sensory data. To obtain high-quality labels, we introduce *Active Sensing Motor Policies*. These policies implement principles of active learning, selecting motor skills that enhance the estimation of environmental physical parameters. The resulting vision model supports real-time pixel-wise estimation of task-agnostic physical properties directly from RGB images. Leveraging a large pretrained vision model, we demonstrate robust generalization in image space, enabling path planning from satellite imagery using only ground camera images for training. Our method improves the adaptability of robot navigation across diverse terrains and conditions. [1]

Keywords: Robot Vision, Self-Supervised Learning, Locomotion

1 Introduction

In recent years, legged locomotion controllers have exhibited remarkable stability and control across a wide range of terrains such as pavement, grass, sand, ice, slopes, and stairs [1, 2, 3, 4, 5]. Despite these advancements, performance metrics such as energy efficiency, safety, and motion characteristics remain variable across different terrains. This variability underscores the necessity for nuanced terrain perception to enable efficient and safe path planning. While geometric information may be incorporated directly into control decisions through sim-to-real training on heightmaps or depth images [3, 5], such approaches discard the semantic terrain information conveyed by color and texture.

Because today's simulators do not capture the diversity of real-world relationships between terrain properties and visual appearance, previous works have facilitated semantics-aware locomotion by

[1] Project website at https://sites.google.com/view/look-and-feel-loco/home

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

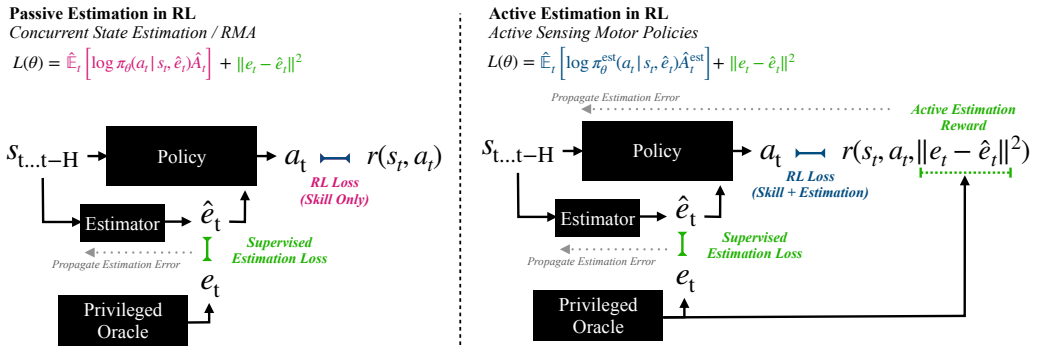


Figure 2: **Active Sensing Motor Policies optimize for estimation.** Unlike passive methods (left) which estimate the state only to the extent that it is observable as a byproduct of control relevance, Active Sensing Motor Policies (right) directly incentivize improved state estimation through the advantage function. This introduces an element of active sensing to the motor skills, which improves the estimate quality.

learning about traversability from real-world experience [6, 7, 8, 9]. We propose two innovations in this methodology. First, we explicitly estimate the terrain physics parameters (i.e. friction coefficient) rather than a traversability metric. This allows us flexibility at evaluation time to reuse the perception module with different controllers or under operating conditions that give the robot a different traversal capability. Second, instead of passively estimating the terrain properties during locomotion [1, 2, 4, 10], we propose training a separate data collection policy that pursues terrain property estimation as part of its task. This *Active Sensing Motor Policy* (ASMP) discovers new locomotion behavior, namely dragging the feet on the ground to better estimate friction, which improves the informativeness of its proprioceptive data. Using this improved data as self-supervision facilitates training a more accurate visual perception module that can generate more efficient plans for different operating states, like dragging heavy objects across the terrain.

Recent advancements in large vision models have provided increasingly general solutions to segmentation [11, 12] and feature extraction [13, 14]. Building upon this progress, our work grounds a pre-trained backbone of self-supervised vision features in the physics governing robot performance. Despite being trained solely with images captured from an onboard camera, our resulting model can interpret and predict terrain semantics from out-of-distribution images. In addition to local path planning from onboard cameras, this lets us perform global planning in satellite imagery or from drone images in a drone-quadruped teaming scenario.

2 Method

Our approach consists of the following stages, which are also illustrated graphically in Figure 1:

1. We train an *Active Sensing Motor Policy* (ASMP) to provide enhanced estimates of the environment dynamics parameters e_t from the proprioceptive sensor history. In our experiments, e_t is the ground friction coefficient. (Section 2.1)

2. Using sparse labels of e_t recorded from real-world traversals of the robot, we learn a function, $\hat{e} = f(\mathbf{I})$, that predicts the per-pixel value of e_t for a given image $\mathbf{I}$. We use pre-trained DINOv2 [14] to facilitate this training. (Section 2.2)

3. To facilitate planning for specific tasks, we create terrains with the same e_t values in simulation. We obtain a cost map for planning by performing rollouts in simulation to measure a cost function $C(e_k)$ that relates dynamics parameters to performance. We construct a separate cost function for each task. (Section 2.3)

4. Combining (2-3), we compute cost maps directly from RGB images and use them for path planning. (Section 2.4)

2.1 Active Sensing Motor Policies: Learning Whole-Body Active Estimation

In learning control policies under partial observations, it is commonplace to train with an implicit [1, 2, 4] or explicit [10] incentive to form representations within the policy network that correspond to the unobserved domain parameters. Consider the concurrent state estimation framework of Ji et al. [10], under which a state estimation network is trained simultaneously with the policy network to predict the unobserved parameters. The predictions of the state estimation network are concatenated with the rest of the observation to construct the policy network input. This approach optimizes a two-part loss consisting of the standard policy objective and the state estimation error:

$$L(\theta, \theta') = \hat{\mathbb{E}}_t \left[\log \pi_\theta(a_t | s_t, \hat{e}_t) \hat{A}_t \right] + \| e_t - \hat{e}_{\theta'}(s_t) \|^2$$

This has been empirically shown to yield better policy performance in environments with domain randomization or unobserved state variables [10].

However, in this formulation, the estimation error is used to update the state estimator weights θ', but not the policy weights θ. This does not incentivize the *policy* to adjust its actions to improve estimation performance beyond what is required for control. Typically, this is no problem because it allows the policy to maximize its performance at the current control task. However, our approach uses the proprioceptive policy to provide sparse labels for training a visual perception module, which we then use with new control tasks. Therefore, to obtain the most accurate perception module, we would like a mechanism to improve the state estimate quality of the proprioceptive data collection policy as much as possible by adapting its behavior. To this end, we propose *Active Sensing Motor Policies* in which the policy π^{est} is trained with an additional *estimation reward*: $r_{\text{est}} = c \cdot \exp\left(\|e - \hat{e}\|^2\right)$. Figure 2 illustrates the policy architecture.

In practice, we observe that an Active Sensing Motor Policy that is rewarded for estimating the ground friction coefficient will – intuitively – slide one foot along the ground or swipe it vigorously to improve the friction coefficient observability in the state history.

2.2 Grounding Self-Supervised Visual Features in Physics from Real-world Experience

We collect paired proprioceptive and vision data from the state estimation policy in the real world in order to learn about the relationship between visual appearance and terrain physics. Specifically, we collect data of the form $(\mathbf{I}, \hat{e}, \mathbf{x})_t$ where $\mathbf{I}$ is a camera image, $\hat{e}$ are the estimated dynamics parameters, and $\mathbf{x}$ is the position and orientation of the robot in a fixed reference frame. We obtain $\mathbf{x}$ by training an extra head of our learned state estimator to predict the displacement $\Delta\mathbf{x}$ at each timestep, and then integrating the estimated displacements. The integrated estimates $\mathbf{x}$ will drift over time, but we will only use them to label nearby points so they are accurate enough for our purposes. This alleviates the need for a separate odometry algorithm to estimate the robot's state.

Using the camera intrinsic and extrinsic transform, we project the measurement points $(\hat{e}, \mathbf{x})_t$ into each camera image frame [6]. We restrict the points to those greater than $1\,\mathrm{m}$ and less than $5\,\mathrm{m}$ from the robot along the traversal path so that they are neither too far away to see nor so close as to be obstructed from view by the robot's own body. After the measurement points are projected, every RGB frame $\mathbf{I}_t$ has a paired sparse measurement image $\mathbf{I}^e_t$, which contains labels of e_t for pixels of $\mathbf{I}_t$ that depict traversed terrain.

For each RGB frame $\mathbf{I}_t$, we use DINOv2 [14] to compute a dense feature mask. We use the smallest distilled model (ViT-S) so as to enable real-time mobile inference. Similar to the procedure that Oquab et al. [14] used for depth estimation, we discretize the labels $\hat{e}_t$ into 20 bins and train a single linear layer with crossentropy loss where the inputs are the features of one patch and the outputs are the logits of the patch's $\hat{e}_t$ label from proprioception.

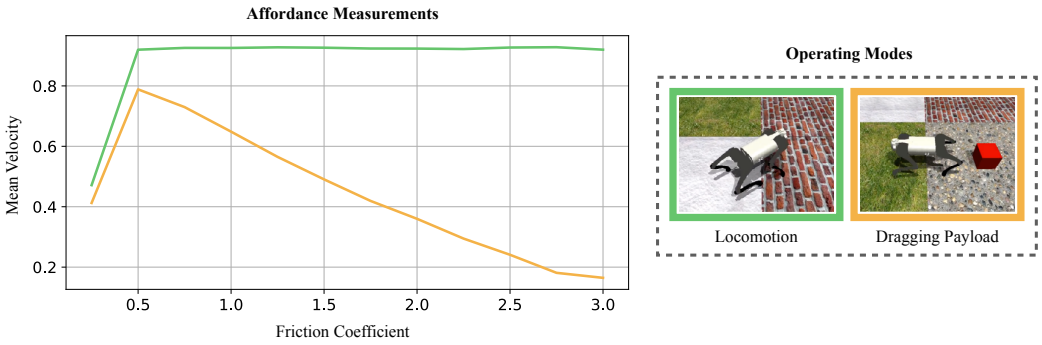


Figure 3: Simulation-based relationship between a robot's operating mode, locomotion affordance at a target velocity ($1\,\mathrm{m/s}$), and terrain friction. In free locomotion, the controller maintains the target velocity across a range of friction coefficients, except in low friction. However, when dragging a weighted box, performance decreases as friction increases.

2.3 Connecting Physics Parameters to Affordances

The impact of terrain properties on robot performance is task dependent: for example, a robot carrying an object may face distinct constraints that inhibit its traversal on some terrains as compared to a robot without any payload. In order to use our vision module for planning, we must establish a mapping between terrain properties and robot performance for each task. We propose a simple procedure for extracting a task cost function from simulated data to demonstrate that our perception module can be useful in planning for multiple tasks, which we refer to as "operating modes". We sample simulated terrains with a variety of terrain properties e_t and command a locomotion policy from prior work [15] to walk forward at $1\,\mathrm{m/s}$. We record the actual resulting velocity achieved on each terrain. We evaluate the mean realized velocity for multiple operating modes: (1) locomotion, (2) payload dragging. For each operating mode, we construct a cost function as the average time spent traversing one meter of a given terrain. Minimizing this cost function during path planning will yield an estimated shortest-time path.

2.4 Integrated Affordance-Aware Path Planning from Vision

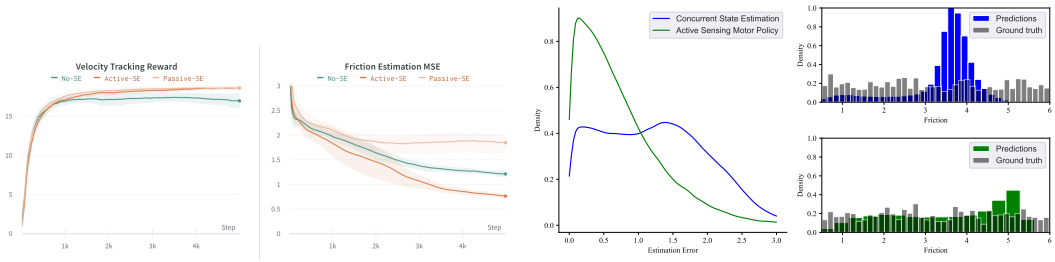
Our perception module (Section 2.2) can run in real-time using onboard compute. Although it was trained using equirectangular images stabilized by a postprocessing step, the resulting pixel-wise friction estimator can be evaluated in images from other cameras including the robot's onboard fisheye camera and an overhead drone or satellite image.

One possible scenario for carrying ground objects across a long distance is that of a drone-quadruped team or a quadruped with access to satellite imagery. In these cases, we can directly evaluate our grounded vision module in overhead images to obtain a pixel-wise friction mask. Then, considering the robot's operating mode, we compute the cost associated with each pixel using the corresponding cost function determined from simulation (Section 2.3). Given this overhead cost map, we use the A* search algorithm [16] to compute the minimum cost traversal path for the current operating state.

2.5 System Setup

Robot We use the Unitree Go1 robot, a 12-motor quadruped robot stands $40\,\mathrm{cm}$ tall. It has an NVIDIA Jetson Xavier NX processor, which runs the control policy and the vision module. For payload dragging experiments, the robot's body is connected to an empty suitcase using a rope.

360 Camera We use a Insta360 X3 360 action camera mounted on the robot to collect images for training the perception module. This camera provides a 360° field of view. Before the image data



(a) Performance and estimate quality during training. (b) Distribution of friction estimates at convergence.

Figure 4: **Learning active estimation.** Active Sensing Motor Policies (`Active-SE`) automatically learn motor skills (e.g. dragging the feet) that improve observability of the environment properties.

is used for training, we use the Insta360 app to perform image stabilization, which takes about two minutes for data collected from a ten-minute run.

Training Compute We perform policy training, video postprocessing, and vision model training on a desktop computer equipped with an NVIDIA RTX 2080 GPU.

Drone Camera For planning from overhead images, we record terrain videos using a DJI Mini 3, a consumer camera drone.

3 Results

3.1 Interaction among Estimation, Adaptation, and Performance

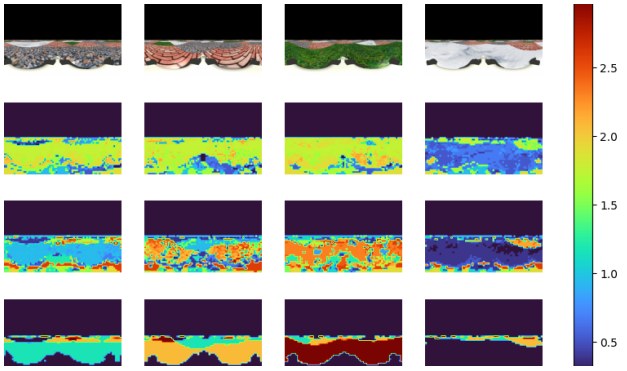
Observing supervised internal state estimates improves proprioceptive locomotion. Confirming the results of Ji et al. [10], we train a state estimation network using supervised learning to predict privileged information (the ground friction coefficient and terrain roughness parameter) from the history of sensory observations. When the policy is allowed to observe the output of this state estimation network (`Passive-SE`), the policy training is more stable and results in a more performant final policy than when the state estimate is not observed (`No-SE`) (Figure 4).

Observing passive state estimates can degrade the system observability. We analyze the error distribution of the learned state estimator in `Passive-SE` and `No-SE` policies (Figure 4). It may be surprising that the friction estimation error of the more-performant `Passive-SE` policy is *worse* than that of the less-performant `No-SE` policy. We propose a mechanistic explanation for this behavior: Supposing some irreducible sensor noise, two terrains of different frictions will only be distinguishable if they make the robot slip in sufficiently different ways. However, a control policy with better adaptive facility is more likely to avoid slip across a wide range of ground frictions. Therefore, in the more adaptive policy, slip will occur less intensely, and as a result, the observability of the ground friction coefficient will degrade.

Our method, ASMP, produces the best privileged state observability. We train an active sensing motor policy (`Active-SE`) to intentionally measure the friction as described in Section 2.1. (The full reward function for each policy we trained is provided in the appendix.) We find that the `Active-SE` policy provides the most accurate friction estimates among the three architectures (Figure 4). Therefore, as we will further show, it is the only policy suitable for supervising a task-agnostic physical grounding for vision.

3.2 Learning to See Friction

Evaluation in Simulated Environment. We collect five minutes of simulated data on four terrains: ice, gravel, brick, and grass, assigning them arbitrary friction coefficients of $\mu = \{0.25, 1.17, 2.08, 3.0\}$ respectively. Figure 5 compares the resulting visual perception module learned from the policies performing passive vs. active estimation. Qualitatively, the vision module



(a) Four example frames (top) and predictions (second, third row) from the simulated equirectangular camera. The model trained with passive proprioceptive sensing (second row) does not distinguish terrains with higher friction. The model trained with active proprioceptive sensing (third row) more closely matches the ground truth (bottom row).

Passive-SE	Active-SE
1.23	0.94

(b) RMSE for visual friction prediction across five minutes of simulated test data. Active Sensing Motor Policies enable more accurate perception.

Figure 5: **Friction inference from RGB images.** We collect one minute of simulated data from policies trained with and without active state estimation and compare the resulting visual inference result against the ground truth.

learned from passive data learns to see ice but fails to distinguish between higher-friction terrains (gravel, brick, and grass). This makes sense as Figure 3 shows that frictions in this range have less influence on the performance of locomotion. In contrast, the vision module trained on data from our Active Sensing Motor Policy correctly learns to distinguish all four terrains. Quantitatively, ASMP results in lower dense prediction loss on images from a held-out test trajectory (Figure 5).

Real-world Training. We collect 15 minutes of real-world traversal data spanning diverse terrains: grass, gravel, dirt, and two types of pavement. Following the procedure in 2.2, we project the traversed points into the corresponding camera images and train a linear head on top of DINOv2 [14] to predict the terrain friction estimate for each traversed patch. The resulting dense predictions may be viewed on the project website; they are generally in agreement with the proprioceptive labels and consistent across regions of similar terrain.

3.3 Integrated Planning

Cost Function Evaluation. We define a cost metric for the locomotion policy from [15] as the distance traveled per second when commanded with a speed of $1.0\,\mathrm{m/s}$. We evaluate this metric in simulation by averaging the performance of 50 agents simulated in parallel for a total of $20\,\mathrm{s}$ on terrains of different friction coefficients ranging from a lower limit of $\mu = 0.25$ to an upper limit of $\mu = 3.0$. This procedure is performed once with the robot in nominal locomotion and again with the robot dragging a $1.0\,\mathrm{kg}$ payload. Figure 3 shows the measured result; both tasks yield poor performance on extremely slippery terrain, but on higher terrains, the robot dragging a payload slows down while the free-moving robot adapts to maintain velocity. Given knowledge of the ground's physical properties, this motivates a difference in high-level navigation decisions between the two tasks.

Path Planning and Execution. We plan paths for locomotion and payload dragging and execute them via teleoperation to evaluate whether the predicted preferences hold true in the real world. We fly a drone over the same environment where the vision model was trained and choose a bird's-eye-view image that includes grass and pavement. We estimate the friction of each pixel and from this we compute the associated cost for locomotion and payload dragging. Then we use A* search to compute optimal paths. The optimized paths and traversal result are shown in Figure 6. In agreement with the planning result, it is preferred to remain on the sidewalk while dragging the payload and cut directly across the grass when in free locomotion.

4 Related Work

Self-supervised traversability estimation has been demonstrated as viable in the context of navigation by both wheeled and legged robots. Some works have focused on the direct estimation of a traversability metric, a scalar value quantifying the cost of traversing a particular terrain, with reference to variables such as velocity tracking error [7, 8, 17]. These approaches are specialized to individual robots and their traversal capability at the time of data collection, implying that a change in system dynamics or control strategy may necessitate repeated data collection and the development of a new vision module.

Other works have demonstrated self-supervised terrain segmentation from proprioceptive data [18, 6, 19]. Wu et al. [18] demonstrated that proprioceptive data from a C-shaped leg equipped with tactile sensors may be sufficient to classify different terrains. Wellhausen et al. [6] gathered supervision from the dominant features of a six-axis force-torque foot sensor during traversal and trained a model to densely predict a "ground reaction score" from RGB images to be used for planning. Łysakowski et al. [19] also demonstrated that terrain classification from proprioceptive readings can be performed in an unsupervised manner on a full-scale quadruped and showed that this information can be used as an additional signal to improve localization. Our work differs from these in that (1) we do not use any dedicated sensor in the foot, but predict the terrain properties using only standard sensors of the robot's egomotion, and (2) thanks to our Active Sensing Motor Policies, we can directly predict the terrain properties instead of a proxy score, which allows us to compute the cost function in simulation for multiple scenarios as in Section 2.3.

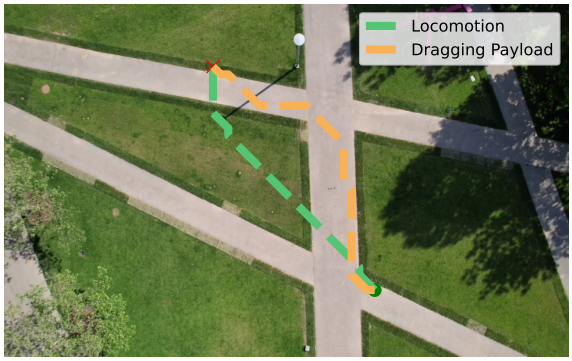
Another possibility is to directly optimize the parameters of a locomotion controller in response to semantic visual information to improve performance for a single task [20, 9]. Loquercio et al. [20] learned to predict the future latent state of RMA [2] from a front-facing camera image to improve low-level control performance in stair climbing. While their approach of predicting the future latent from the entire image improved the low-level adaptation of the robot, its visual representation is specialized to the latent of a single motor policy, so it cannot be reused for new policies or operating states, and predicting the next latent is only meaningful for egocentric images, so it cannot be used for novel viewpoints as in drone-quadruped teaming or planning from satellite imagery. Yang et al. [9] trained a semantic visual perception module for legged quadrupeds using human supervision rather than proprioceptive supervision. The resulting system adapted a scalar quantity in response to different terrains, controlling velocity and gait. This system relies on a human operator to intuitively predict the terrain properties during the demonstration, and its representation shares the specialized nature of a traversability metric.

Most similar to our approach are those works which developed vision modules that visually estimate the geometry or contact properties of the terrain and then compute the traversal cost from these metrics [21, 22]. However, these works have primarily dealt with wheeled robots, which have relatively simple control strategies and limited controller design spaces. Consequently, the question of selecting a controller to gather the most informative label data for such vision systems has not been directly addressed.

Active perception, in which a robot agent actively selects actions to increase environmental observability, offers a framework for understanding this problem. This approach has been long applied to vision systems [23, 24], and more recently, it has been extended to include physical interaction [25, 26, 27]. Thus, it presents a promising avenue to address the controller selection issue in labeling vision with proprioception for legged robots.

5 Discussion of Limitations

Our approach assumes a one-to-one mapping between terrain appearance and physics. In practice, identical looking terrains could have different physical characteristics. Future work could explore representing uncertainty or performing fast online adaptation of the estimates to the current environment based on new proprioceptive information.



Operating Mode	Metric	Cross Grass	Stay on Sidewalk
Dragging Payload	Time (s)	48 ± 1	45 ± 1
Locomotion	Time (s)	23 ± 1	26 ± 0

Figure 6: **Path Planning in Overhead Images.** We exploit the generalization of the learned vision module to plan navigation in overhead images of terrain. The vision module is only trained using first-person views from the robot, but robustness of the pretrained vision model enables inference of physical properties with a different camera model and viewing pose. We teleoperate the robot across both planned paths in both locomotion modes. The preference among paths in the real world matches the planning result from our pipeline.

Our method is fast and lightweight enough to deploy on-robot training and inference. However, processing the panoramic camera footage for training requires an export step using software not available on Linux. In the future, we could pursue onboard, real-time training using streaming images from a different camera.

We have only fully implemented the pipeline for friction estimation on terrains of varied roughness. We don't know what will happen if we expand the range of terrain properties to e.g. softness or compliance. One possibility is that we would need a separate probing policy for each property if they each require different movements to measure accurately. Our approach also requires all relevant terrain properties to be represented in simulation during training. As an example, granular terrains, which we never simulate, can be sensed as low-friction due to their similar effect, and this may be sufficient to predict performance for some tasks. But, if the robot were targeting some very granularity-dependent task like scooping up the terrain, we would need to add granular terrains to the simulated model and train a new active sensing skill that specifically disentangles granularity from friction.

The base locomotion controller we used does not incorporate geometric perception unlike some recent works [3, 5]. We specifically aimed to study semantic perception. In the future, we believe our method can be integrated with existing work on geometry-perceptive locomotion without major modification.

6 Conclusion

It is promising to learn visual representations for robotics directly from proprioceptive supervision, which can provide useful information about the relationship between appearance and texture. In this work, we exposed that the quality of proprioceptive supervision can be strongly influenced by the style of the motor policy acquired through reinforcement learning. We proposed a novel technique, Active Sensing Motor Policies, and show that it improves the proprioceptive supervision quality and the corresponding performance of a grounded vision module that is resusable for new sensor configurations and physical tasks.

References

[1] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Sci. Robot.*, 5(47):eabc5986, Oct. 2020. doi:10.1126/scirobotics.abc5986.

[2] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid motor adaptation for legged robots. In *Proc. Robot.: Sci. and Syst. (RSS)*, June 2021.

[3] T. Miki, J. Lee, J. Hwanbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Sci. Robot.*, 7(62):abk2822, Jan. 2022. doi:10.1126/scirobotics.abk2822.

[4] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid locomotion via reinforcement learning. In *Proc. Robot.: Sci. and Syst. (RSS)*, June 2022.

[5] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Proc. Conf. Robot Learn. (CoRL)*, Auckland, New Zealand, Dec. 2022.

[6] 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 Robotics and Automation Letters*, 4(2):1509–1516, 2019.

[7] M. G. Castro, S. Triest, W. Wang, J. M. Gregory, F. Sanchez, J. G. Rogers III, and S. Scherer. How does it feel? self-supervised costmap learning for off-road vehicle traversability. *arXiv preprint arXiv:2209.10788*, 2022.

[8] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter. Fast traversability estimation for wild visual navigation. *arXiv preprint arXiv:2305.08510*, 2023.

[9] Y. Yang, X. Meng, W. Yu, T. Zhang, J. Tan, and B. Boots. Learning semantics-aware locomotion skills from human demonstration. In *Conference on Robot Learning*, pages 2205–2214. PMLR, 2023.

[10] G. Ji, J. Mun, H. Kim, and J. Hwangbo. Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion. *IEEE Robot. Automat. Lett. (RA-L)*, 7 (2):4630 – 4637, Apr. 2022. doi:10.1109/LRA.2022.3151396.

[11] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[12] J. Xu, S. Liu, A. Vahdat, W. Byeon, X. Wang, and S. De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. *arXiv preprint arXiv:2303.04803*, 2023.

[13] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[14] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[15] G. B. Margolis and P. Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. In *Proc. Conf. Robot Learn. (CoRL)*, Auckland, New Zealand, Dec. 2022.

[16] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[17] Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, and D. Pathak. Coupling vision and proprioception for navigation of legged robots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17273–17283, 2022.

[18] X. A. Wu, T. M. Huh, R. Mukherjee, and M. Cutkosky. Integrated ground reaction force sensing and terrain classification for small legged robots. *IEEE Robotics and Automation Letters*, 1(2):1125–1132, 2016.

[19] M. Łysakowski, M. R. Nowicki, R. Buchanan, M. Camurri, M. Fallon, and K. Walas. Unsupervised learning of terrain representations for haptic monte carlo localization. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4642–4648. IEEE, 2022.

[20] A. Loquercio, A. Kumar, and J. Malik. Learning visual locomotion with cross-modal supervision. *arXiv preprint arXiv:2211.03785*, 2022.

[21] A. J. Sathyamoorthy, K. Weerakoon, T. Guan, J. Liang, and D. Manocha. Terrapn: Unstructured terrain navigation using online self-supervised learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7197–7204. IEEE, 2022.

[22] X. Meng, N. Hatch, A. Lambert, A. Li, N. Wagener, M. Schmittle, J. Lee, W. Yuan, Z. Chen, S. Deng, et al. Terrainnet: Visual modeling of complex terrain for high-speed, off-road navigation. *arXiv preprint arXiv:2303.15771*, 2023.

[23] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

[24] S. K. Ramakrishnan, D. Jayaraman, and K. Grauman. Emergence of exploratory look-around behaviors through active observation completion. *Science Robotics*, 4(30):eaaw6326, 2019.

[25] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, 2017.

[26] H. Van Hoof, O. Kroemer, H. B. Amor, and J. Peters. Maximally informative interaction learning for scene exploration. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5152–5158. IEEE, 2012.

[27] V. Chu, I. McMahon, L. Riano, C. G. McDonald, Q. He, J. M. Perez-Tejada, M. Arrigo, T. Darrell, and K. J. Kuchenbecker. Robotic learning of haptic adjectives through physical interaction. *Robotics and Autonomous Systems*, 63:279–292, 2015.

Figure 7

A Reward Function for Policy Training

B Hyperparameters and Architecture for Vision Module

C Evaluation on Out-of-Distribution Images

D Path Planning Procedure