

Learning to See Physical Properties with Active Sensing Motor Policies

Gabriel B. Margolis Xiang Fu Yandong Ji Pulkit Agrawal

Improbable AI Lab, Massachusetts Institute of Technology

<https://gmargo11.github.io/active-sensing-loco>

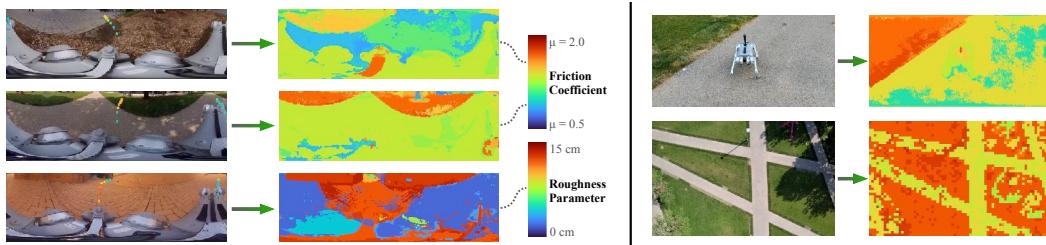


Figure 1: **Learning to see how terrains feel.** *Left:* Sparse training labels acquired from proprioceptive traversal distinguish dirt (top), grass (middle), and stairs (bottom) to supervise dense perception of physical properties. *Right:* Prediction from novel viewpoints facilitates locomotion planning.

Abstract: Knowledge of terrain’s physical properties inferred from color images can aid in making efficient robotic locomotion plans. However, unlike image classification, it is unintuitive for humans to label image patches with physical properties. Without labeled data, building a vision system that takes as input the observed terrain and predicts physical properties remains challenging. We present a method that overcomes this challenge by self-supervised labeling of images captured by robots during real-world traversal with physical property estimators trained in simulation. To ensure accurate labeling, we introduce *Active Sensing Motor Policies* (ASMP), which are trained to explore locomotion behaviors that increase the accuracy of estimating physical parameters. For instance, the quadruped robot learns to swipe its foot against the ground to estimate the friction coefficient accurately. We show that the visual system trained with a small amount of real-world traversal data accurately predicts physical parameters. The trained system is robust and works even with overhead images captured by a drone despite being trained on data collected by cameras attached to a quadruped robot walking on the ground.

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, 6, 7, 8]. State-of-the-art approaches using sim-to-real learning primarily rely on proprioception and depth sensing to perceive obstacles and terrain [5, 7, 8, 9, 10, 11, 12, 13, 14, 15]. These approaches discard valuable information about the terrain’s material properties beyond geometry, such as slip, softness, etc., conveyed by color images. A primary reason for this choice is that sim-to-real transfer has been shown to work with depth images [5, 7, 10], but it remains unclear how well the transfer will work with color or RGB images. To utilize information beyond geometry, some works learn to predict task performance or task-relevant properties (e.g., traversability) from color images using data collected in the real world [16, 17, 18, 19, 20]. However, the terrain property predictors learned in prior works are task- or policy-specific, which limits their applicability to new tasks.

To perceive a multipurpose representation of the terrain, we propose predicting the terrain’s physical properties (e.g., friction, roughness) that are invariant to the policy and task. Perceiving the

physical parameters makes it possible to create a *digital twin* of the terrain in front of the robot and use *simulation* to estimate the cost map for a new task (e.g., dragging a payload) or objective (e.g., preference for speed or energy efficiency). One way to generate the cost map is to collect many rollouts of the policy in simulation and label each location with the value function associated with the new objective. This costmap can be used to plan a trajectory that can be executed in reality.

The main obstacle in this approach is collecting a dataset of terrain images labeled with associated physical properties. A natural way of collecting labels is to traverse the terrain and proprioceptively estimate its physical parameters with a neural network supervised by the ground truth labels available in simulation [4, 5, 21]. We discovered that the estimates obtained in this way can be imprecise because the locomotion behavior often makes the terrain properties hard to predict. Therefore, unlike prior works in terrain perception that predict the terrain characteristics from passive data [17, 18, 19, 20], we propose training a specialized data collection policy that directly optimizes for terrain sensing. This *Active Sensing Motor Policy* (ASMP) learns emergent locomotion behavior, such as dragging the feet on the ground to better estimate friction, and improves the informativeness of its proprioceptive traversals.

We use the improved data obtained through ASMP as self-supervision to learn a visual perception module that predicts terrain material properties (Figure 1). The same model can inform efficient plans for nominal locomotion and for dragging objects by considering the impact of terrain properties on traversal cost. Because the robot is low to the ground, its onboard cameras only provide enough range for local planning. Although our model is trained only with data collected by the robot, it can also be evaluated to predict terrain properties using images from various viewpoints. Therefore, we also consider data from a teamed drone that flies above the legged robot and show that it successfully informs traversal from an extended view of the environment.

2 Method

Our approach consists of the following stages, which are also illustrated graphically in Figure 2:

- 1. Active Sensing:** We estimate the terrain dynamics parameter, e_t , from the proprioceptive sensor history during an initial blind traversal. Our *Active Sensing Motor Policy* (ASMP) crucially provides better-calibrated estimates than the baseline policy. In our experiments, the estimated parameter e_t is the ground friction coefficient, the ground roughness magnitude, or both. (Section 2.1)
- 2. Self-Supervised Vision Learning:** Using labels of e_t recorded from the real-world traversal 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} . The labels for training are only available at the pixels corresponding to the places the robot traversed, but the resulting model can be queried to predict the terrain parameter at any pixel. (Section 2.2)
- 3. Cost Function Learning:** To inform planning, we learn cost functions that relate the terrain dynamics to various performance metrics. First, we create terrains with a range of e_t values in simulation. Then, we perform rollouts in simulation to measure a cost function $C(e_k)$ that relates dynamics parameters to performance. We learn a separate cost function for each task. (Section 2.3)
- 4. Dynamics-Aware Path Planning:** Combining (2-3), we compute cost maps directly from color images and use them for path planning. (Section 2.4)

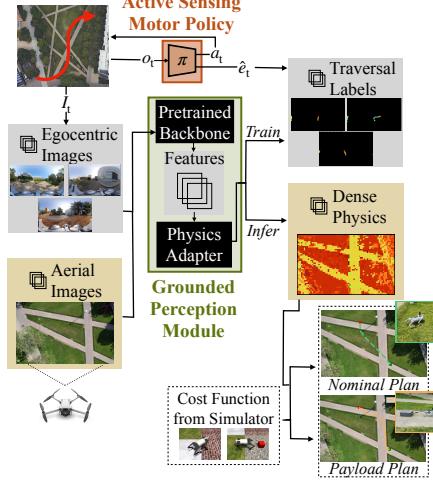


Figure 2: **Active self-supervision.** We propose learning an optimized gait for collecting informative proprioceptive terrain labels that supervise training for a vision module, which can be used for navigation planning with new tasks and views.

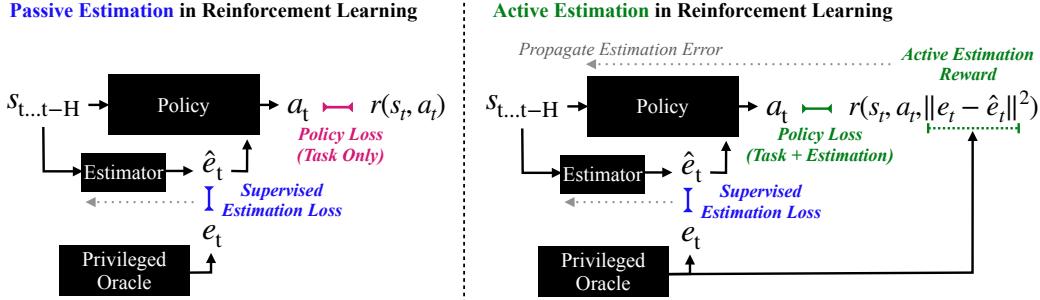


Figure 3: **Active Sensing Motor Policies optimize for estimation.** When training in simulation, an oracle can provide privileged state information e_t that cannot be directly measured on the real robot, such as the present roughness and friction of the terrain. Prior works learn an estimator network to make predictions \hat{e}_t from the history of sensor readings that are available on the real robot $s_{t...t-H}$ (left). We propose additionally optimizing the policy for estimation (right). This incentivizes information-gathering behaviors, like intentionally swiping the robot’s foot during legged locomotion to estimate the terrain properties more accurately.

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, 3, 5, 8] or explicit [4, 22] incentive to form representations within the policy network that correspond to the unobserved dynamics parameters. Consider the concurrent state estimation framework of Ji et al. [4], where 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: $L(\theta, \theta') = \hat{\mathbb{E}}_t[\log \pi_\theta(a_t | s_t, \hat{e}_t) \hat{A}_t] + \|e_t - \hat{e}_{\theta'}(s_t)\|^2$. This has been empirically shown to yield better policy performance in environments with randomized dynamics or partial observations [4, 21].

In the formulation above, 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 end goal is to use the output of the state estimator to train a visual perception module that may be reused with other controllers and tasks. To support this, the labels should be as accurate as possible even when that is not necessary for control. 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 the policy’s behavior. To this end, we propose *Active Sensing Motor Policies* in which the policy π^{est} is trained with an additional *estimation reward*: $r_{\text{est}} = c \cdot \exp(-\|\hat{e} - e\|^2)$. Figure 3 illustrates the policy architecture. In practice, we observe that an Active Sensing Motor Policy that is rewarded for estimating the ground friction coefficient slides one foot along the ground or swipes it vigorously to improve the friction coefficient observability in the state history.

2.2 Grounding 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 additional 2D output of the final MLP layer in our learned state estimator to predict the displacement in the ground plane of the base from its location at the previous timestep, $\Delta\mathbf{x}$, and then integrate the estimated displacements. The integrated estimates \mathbf{x} will drift over time, but we

will only rely on them over a short time window. 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 relative positions of the robot in the past and future into each camera image frame. We restrict the positions to those between 1 m and 5 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. We label each of the projected robot positions with the estimated dynamics parameters \hat{e} that the robot felt when it walked there. This yields a corresponding label image I_t^e for each color image I where the traversed pixels are labeled with their measured dynamics.

For each color frame I_t , we use the pretrained convolutional backbone [23] to compute a dense feature map. Similar to the procedure that Oquab et al. [24] used for depth estimation, we discretize the labels \hat{e}_t into 20 bins and train a single linear layer with cross-entropy 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.

2.3 Cost Function Learning: Connecting Physics Parameters to Affordances

The impact of terrain properties on robot performance is task-dependent: for example, a robot dragging an object may face distinct constraints that inhibit its traversal on some terrains, compared to a robot without any payload. 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 [22] to walk forward at 1 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. We construct a cost function for each operating mode 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. While we focus on time-optimal payload dragging as an example, (1, 2) could be any combination of task and metric as long as their relation to terrain properties can be evaluated in simulation.

2.4 Integrated Dynamics-Aware Path Planning from Vision

Our perception module (Section 2.2) runs in real-time (2 Hz) using onboard compute. Although it was trained using images from a 360-degree camera, 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. This is useful because the perception module can remain useful when deployed on a new robot or evaluated from a new viewpoint.

One possible scenario for carrying ground objects across a long distance is that of a drone-quadruped team. In this case, 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 [25] to compute the minimum cost traversal path for the current operating state.

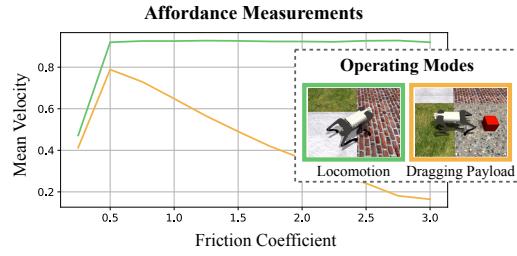
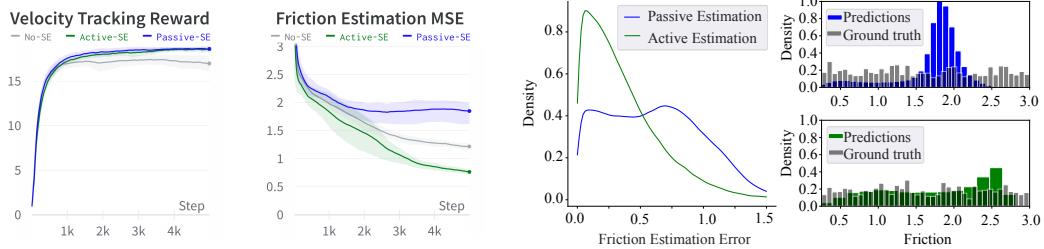


Figure 4: **Locomotion affordances.** We measure the dependence of locomotion performance (1 m/s) on terrain friction in two different operating modes. In free locomotion, the controller maintains the target velocity across a range of friction coefficients, except for the lowest friction. In contrast, when dragging a weighted box, the robot slows down as the terrain friction increases.



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

Figure 5: **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.

2.5 System Setup

Robot: We use the Unitree Go1 robot, a 12-motor quadruped robot standing 40 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 an 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 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. Affirming the results of Ji et al. [4], 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 5).

Observing passive state estimates can degrade the state observability. We compute the error distribution of the learned state estimator in Passive-SE and No-SE policies (Figure 5). It may be surprising that the friction estimation error of the more-performant Passive-SE policy is *higher* than that of the less-performant No-SE policy. We suggest an explanation for this: 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 a better adaptive facility is more likely to avoid slipping across a wide range of ground frictions. Because slip occurs less frequently in the more adaptive policy, the observability of the ground friction coefficient degrades.

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 5). Therefore, as we will further show, it is the superior policy for supervising a task-agnostic physical grounding for vision.

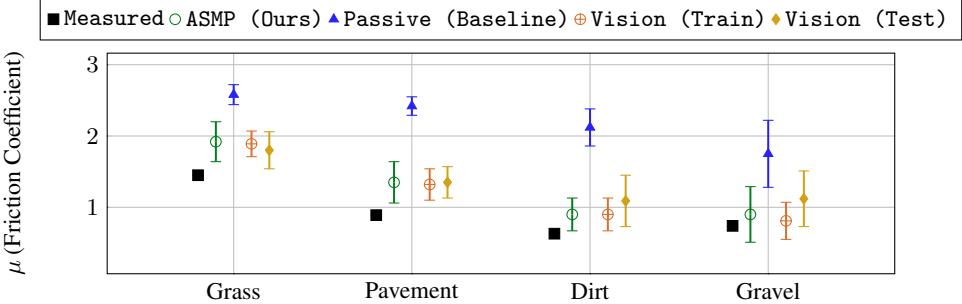


Figure 6: **Real-world friction sensing performance with proprioception and vision.** Measured values are directly measured by a dynamometer. The predictions from our proposed ASMP (Ours) agree better with the dynamometer measurements than the baseline Passive (Baseline). Vision (Train) shows the generalization of visual prediction to un-traversed patches in the training images from the onboard camera; (Vision (Test)) shows the generalization to unseen patches and viewpoints by evaluating on drone footage. We use manual segmentation maps (Appendix Figure 9) to match pixel predictions to terrains. Error bars indicate one standard deviation.

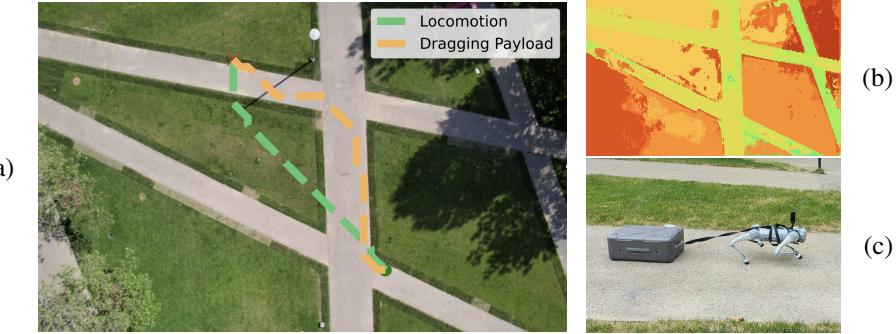
3.2 Learning to See Physical Properties

Real-world Evaluation. We collect fifteen minutes of real-world traversal data spanning diverse terrains: grass, gravel, dirt, pavement, and stairs. Following the procedure in Section 2.2, we project the traversed points into the corresponding camera images and train a linear head on top of a convolutional backbone pretrained for segmentation [23] to predict the terrain friction and roughness estimate for each traversed patch. To evaluate estimation performance in the real world, we manually label image segments in a subset of train and test images containing grass, pavement, dirt, or gravel and compute the distribution of proprioceptive and visual friction predictions for each (Figure 6). We measure a ground truth friction value for each terrain using a dynamometer by measuring the weight of a payload made of robot foot material and its drag force. The proprioceptive estimates from ASMP are much closer to the dynamometer measurements than the estimates from the passive baseline. They do not match perfectly, suggesting a small but measurable sim-to-real gap in the robot dynamics or terrain modeling. They agree with the dynamometer measurements on the ordering of terrains from most to least slippery. The grounded vision module is close to the distribution of proprioceptive estimates for both train and test images, with increased variance in test images.

3.3 Integrated Planning

Cost Function Evaluation. We define a cost metric for the locomotion policy from [22] as the distance traveled per second when commanded with a speed of 1.0 m/s. We evaluate this metric in simulation by averaging the performance of 50 agents simulated in parallel for 20 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 kg payload. Figure 4 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. Knowledge of the ground’s physical properties 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 7. 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.



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

Figure 7: **Path planning in overhead images.** (a) We use the learned vision module to plan navigation in overhead images of terrain. (b) The vision module is only trained using first-person views from the robot but can infer the terrain friction with a different camera model and viewing pose. (c) We teleoperate the robot across both planned paths in each locomotion mode. The preference among paths in the real world matches the planning result from our pipeline.

4 Related Work

Self-supervised traversability estimation has been studied previously for the navigation of 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 [18, 19, 26]. These approaches are specialized to the robot’s traversal capability at the time of data collection, implying that a change in the policy or task may necessitate repeated data collection to train a new vision module.

Other works have demonstrated self-supervised terrain segmentation from proprioceptive data [17, 27, 28]. Wu et al. [27] demonstrated that proprioceptive data from a C-shaped leg equipped with tactile sensors may be sufficient to classify different terrains. Wellhausen et al. [17] took 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 color images to be used for planning. Łysakowski et al. [28] also demonstrated that terrain classification from proprioceptive readings could be performed unsupervised on a full-scale quadruped and showed that this information could 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 ego-motion, and (2) we 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 predict which locomotion skill to execute from visual information [20, 29]. Loquercio et al. [29] learned to predict the future latent state of the policy from a front-facing camera image to improve low-level control performance in stair climbing. An advantage of their approach is that it does not require the choice of an explicit terrain parameterization, but this comes at the cost that 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. [20] trained a semantic visual perception module for legged quadrupeds using human demonstrations. The resulting system imitated an operator’s response to different terrains, controlling velocity and gait. This relies on a human operator to predict the terrain properties during the demonstration. Other work has learned general navigation through supervised learning on diverse robotic platforms, including legged robots [30, 31, 32]. Training an omni-policy for all robots and environments enables interesting zero-shot generalization,

but the resulting navigation decisions do not account for the impact of embodiment (wheeled/legged) or varied operating conditions (carrying a payload) on traversability.

Several works on wheeled robots visually estimate the geometry or contact properties of the terrain through self-supervision or hand-designed criteria and then compute the traversal cost from these metrics [16, 33, 34, 35, 36, 37, 38, 39, 40]. Wheeled robots have a limited variety of traversal strategies compared to legged robots. Consequently, the question of selecting a locomotion controller to gather the most informative self-supervision data has not been directly addressed. Active perception suggests a solution in which a robot agent optimizes its behavior to sense the environment. This approach has been applied to vision systems [41, 42, 43], and more recently has been extended to include physical interaction [44, 45, 46, 47]. This inspired our approach to the controller selection issue in labeling vision with proprioception for legged robots.

5 Discussion and Limitations

Our work assumes a mapping between the estimated terrain properties and the robot’s performance. Friction affects the slip of the robot’s feet against the ground and the drag force of payloads and other objects, so it is an interesting factor of performance variation for practical locomotion tasks. Of course, expanding the representation of terrain physics to include additional material properties will enable more accurate performance prediction in more diverse tasks. To account for other parameters besides friction that vary in the environment, our framework can be extended to include them. For example, Figure 8 shows that ASMP successfully enhances the accuracy of estimating terrain roughness in addition to friction.

In general, ASMP may be applied for terrain labeling under two conditions: (1) a history of proprioceptive readings is sufficient to infer the parameter of interest, and (2) the parameter of interest can be effectively simulated. If these conditions are not met, a different technique besides ASMP may be necessary to collect training data. Additionally, to train our vision module, we assume terrains with different properties are visually different. If some parameters do not impact the terrain’s visual appearance, learning a vision module of the form we propose for those parameters may be impossible. Future work could explore methods to address this, such as representing uncertainty or performing an online adaptation of the estimates to the current environment based on new proprioceptive information. Finally, our labeling method assumes a mostly visible environment and does not account for occlusions. Future work can combine our system with geometric mapping to alleviate this.

The performance of generalized segmentation models is rapidly improving, and these methods will be effective for distinguishing objects relevant to robot behavior based on multimodal specifications like language descriptions or reference images [23, 24, 48, 49, 50]. However, the prevailing datasets for training these models do not include physical interactions, so they cannot directly predict physical properties. Moreover, the physical properties of terrain can change depending on the conditions; For example, recent rainfall may muddy a grassy field without changing its visual appearance. Therefore, it is a benefit of our method that it is informed by recently collected data from the target environment rather than relying exclusively on offline pretraining.

Estimation Mode	Friction Loss	Rough Loss	Torque Penalty
Passive	1.00	1.00	-0.34
Friction	0.47	1.06	-0.87
Roughness	0.99	0.72	-0.84
Joint Fr.+Ro. Fr.+Ro.	0.49	0.80	-1.18

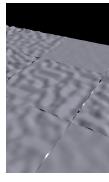


Figure 8: **ASMP for multiple physical parameters.** Friction and roughness estimates are improved by ASMP, even when both parameters are jointly targeted. We report estimation loss for passive estimation (None (Passive)), active estimation of each parameter separately (Friction, Roughness), and active estimation for both parameters in a single policy (Joint Fr.+Ro.). Variation in torque reflects that a change in motor strategy enabled the improved estimation. *Right image:* Terrain with varied roughness parameter.

Acknowledgments

We thank the members of the Improbable AI lab for the helpful discussions and feedback on the paper. We are grateful to MIT Supercloud and the Lincoln Laboratory Supercomputing Center for providing HPC resources. This research was supported by the DARPA Machine Common Sense Program, the MIT-IBM Watson AI Lab, and the National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, <http://iaifi.org/>). We acknowledge support from ONR MURI under grant number N00014-22-1-2740. This research was also sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

Author Contributions

- **Gabriel B. Margolis** ideated, implemented, and evaluated Active Sensing Motor Policies and shared ideation and implementation of the vision module and overall experimental design.
- **Xiang Fu** shared ideation and implementation of vision module and overall experimental design.
- **Yandong Ji** contributed ideas and supported infrastructure development during the project.
- **Pulkit Agrawal** advised the project and contributed to its conceptual development, experimental design, positioning, and writing.

References

- [1] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.
- [2] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid motor adaptation for legged robots. *Robotics: Science and Systems*, 2021.
- [3] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid locomotion via reinforcement learning. *Robotics: Science and Systems*, 2022.
- [4] 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 Robotics and Automation Letters*, 7(2):4630–4637, 2022.
- [5] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022.
- [6] Y. Ji, G. B. Margolis, and P. Agrawal. Dribblebot: Dynamic legged manipulation in the wild. *arXiv preprint arXiv:2304.01159*, 2023.
- [7] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on Robot Learning*, pages 403–415. PMLR, 2023.
- [8] S. Choi, G. Ji, J. Park, H. Kim, J. Mun, J. H. Lee, and J. Hwangbo. Learning quadrupedal locomotion on deformable terrain. *Science Robotics*, 8(74):eade2256, 2023.
- [9] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter. Learning a state representation and navigation in cluttered and dynamic environments. *IEEE Robotics and Automation Letters*, 6(3):5081–5088, 2021.

- [10] G. B. Margolis, T. Chen, K. Paigwar, X. Fu, D. Kim, S. Kim, and P. Agrawal. Learning to jump from pixels. *Conference on Robot Learning*, 2021.
- [11] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang. Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers. *arXiv preprint arXiv:2107.03996*, 2021.
- [12] I. M. A. Nahrendra, B. Yu, and H. Myung. Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5078–5084. IEEE, 2023.
- [13] S. Kareer, N. Yokoyama, D. Batra, S. Ha, and J. Truong. Vinl: Visual navigation and locomotion over obstacles. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2018–2024. IEEE, 2023.
- [14] J. Truong, A. Zitkovich, S. Chernova, D. Batra, T. Zhang, J. Tan, and W. Yu. Indoorsim-to-outdoorreal: Learning to navigate outdoors without any outdoor experience. *arXiv preprint arXiv:2305.01098*, 2023.
- [15] R. Yang, G. Yang, and X. Wang. Neural volumetric memory for visual locomotion control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1430–1440, 2023.
- [16] R. Hadsell, P. Serenat, J. Ben, A. Erkan, J. Han, B. Flepp, U. Muller, and Y. LeCun. Online learning for offroad robots: Using spatial label propagation to learn long-range traversability. In *Proc. of Robotics: Science and Systems (RSS)*, volume 11, page 32. Citeseer, 2007.
- [17] 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.
- [18] 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.
- [19] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter. Fast traversability estimation for wild visual navigation. *Robotics: Science and Systems*, 2023.
- [20] 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.
- [21] W. Yu, J. Tan, C. K. Liu, and G. Turk. Preparing for the unknown: Learning a universal policy with online system identification. *arXiv preprint arXiv:1702.02453*, 2017.
- [22] G. B. Margolis and P. Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. In *Conference on Robot Learning*, pages 22–31. PMLR, 2023.
- [23] Q. Yu, J. He, X. Deng, X. Shen, and L.-C. Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *arXiv preprint arXiv:2308.02487*, 2023.
- [24] M. Oquab, T. Darcret, 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.
- [25] 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.

- [26] 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.
- [27] 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.
- [28] 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.
- [29] A. Loquercio, A. Kumar, and J. Malik. Learning visual locomotion with cross-modal supervision. *arXiv preprint arXiv:2211.03785*, 2022.
- [30] S. Levine and D. Shah. Learning robotic navigation from experience: principles, methods and recent results. *Philosophical Transactions of the Royal Society B*, 378(1869):20210447, 2023.
- [31] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7226–7233. IEEE, 2023.
- [32] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine. Vint: A foundation model for visual navigation. *arXiv preprint arXiv:2306.14846*, 2023.
- [33] R. Hadsell, P. Serenat, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. Muller, and Y. Le-Cun. Learning long-range vision for autonomous off-road driving. *Journal of Field Robotics*, 26(2):120–144, 2009.
- [34] D. Stavens and S. Thrun. A self-supervised terrain roughness estimator for off-road autonomous driving. *arXiv preprint arXiv:1206.6872*, 2012.
- [35] S. Palazzo, D. C. Guastella, L. Cantelli, P. Spadaro, F. Rundo, G. Muscato, D. Giordano, and C. Spampinato. Domain adaptation for outdoor robot traversability estimation from rgb data with safety-preserving loss. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10014–10021. IEEE, 2020.
- [36] H. Lee and W. Chung. A self-training approach-based traversability analysis for mobile robots in urban environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3389–3394. IEEE, 2021.
- [37] X. Xiao, J. Biswas, and P. Stone. Learning inverse kinodynamics for accurate high-speed off-road navigation on unstructured terrain. *IEEE Robotics and Automation Letters*, 6(3):6054–6060, 2021.
- [38] A. Shaban, X. Meng, J. Lee, B. Boots, and D. Fox. Semantic terrain classification for off-road autonomous driving. In *Conference on Robot Learning*, pages 619–629. PMLR, 2022.
- [39] 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.
- [40] 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.
- [41] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

- [42] D. Jayaraman and K. Grauman. Look-ahead before you leap: end-to-end active recognition by forecasting the effect of motion. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part V 14*, pages 489–505. Springer, 2016.
- [43] S. K. Ramakrishnan, D. Jayaraman, and K. Grauman. Emergence of exploratory look-around behaviors through active observation completion. *Science Robotics*, 4(30):eaaw6326, 2019.
- [44] 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.
- [45] 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.
- [46] 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.
- [47] D. Pathak, P. Mahmoudieh, G. Luo, P. Agrawal, D. Chen, Y. Shentu, E. Shelhamer, J. Malik, A. A. Efros, and T. Darrell. Zero-shot visual imitation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2050–2053, 2018.
- [48] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.
- [49] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.
- [50] 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.

Table 1: Reward terms for No-SE, Passive-SE, and Active-SE policy training.

Reward Terms		
Term	Expression	Weight
XY Vel	$\exp\{- \mathbf{v}_{xy} - \mathbf{v}_{xy}^{\text{cmd}} ^2/\sigma_{vxy}\}$	1.0
Yaw Vel	$\exp\{-(\omega_z - \omega_z^{\text{cmd}})^2/\sigma_{\omega z}\}$	0.5
Swing Phase	$[1 - \kappa]\exp\{-\delta_{cf} \mathbf{f}^{\text{foot}} ^2\}$	-4.0
Stance Phase	$\kappa\exp\{-\delta_{cv} \mathbf{v}_{xy}^{\text{foot}} ^2\}$	-4.0
Joint Limits	$\mathbf{1}_{q_i > q_{\max}} \mathbf{1}_{q_i < q_{\min}}$	-10.0
Joint Torque	$ \tau ^2$	-0.0001
Joint Velocity	$ \dot{\mathbf{q}} $	-0.0001
Joint Acceleration	$ \ddot{\mathbf{q}} $	$-2.5e-7$
Hip/Thigh Collision	$\mathbf{1}_{\text{collision}}$	-5.0
Projected Gravity	$ \mathbf{g}_{xy} ^2$	-5.0
Action Smoothing	$ \mathbf{a}_{t-1} - \mathbf{a}_t ^2$	-0.1
Action Smoothing 2	$ \mathbf{a}_{t-2} - 2\mathbf{a}_{t-1} + \mathbf{a}_t ^2$	-0.1
ASMP Bonus	$ \mathbf{e}_t - \hat{\mathbf{e}}_t ^2$	-0.3

Table 2: Notation, observation and action space.

Parameter	Definition	Units	Dimension
<i>Learned Policies</i>			
$\pi_{\text{No-SE}}$	No State Est	-	-
$\pi_{\text{Passive-SE}}$	Passive State Est	-	-
$\pi_{\text{Active-SE}}$	Active State Est	-	-
π_{Loco}	Policy from [22]	-	-
<i>Policy Observation (\mathbf{o})</i>			
\mathbf{q}	Joint Angles	rad	12
$\dot{\mathbf{q}}$	Joint Velocities	rad/s	12
\mathbf{g}	Norm Gravity, Body Frame	m/s ²	3
ψ_t	Body Yaw, Global Frame	rad	1
\mathbf{a}_{t-1}	Previous Action	-	12
θ^{cmd}	Timing Reference [22]	-	4
$\hat{\mathbf{e}}_t$	Estimator Output	-	5
<i>Policy Action (\mathbf{a})</i>			
\mathbf{a}_{des}	Joint Position Targets	rad	12
<i>Other Quantities</i>			
\mathbf{f}^{foot}	Foot Vertical Force	N	1
$\mathbf{v}_{xy}^{\text{foot}}$	Foot xy-Velocity	m/s	1
κ	Desired Contact State	-	1
$\mathbf{v}_{xy}^{\text{cmd}}$	Target x-y Linear Velocity	m/s	2
\mathbf{v}_{xy}	Actual x-y Linear Velocity	m/s	2
ω_z^{cmd}	Target Yaw Velocity	rad/s	1
ω_z	Actual Yaw Velocity	rad/s	1
τ	Joint Torque	N m	1

A Reward Function for Policy Training

We follow the reward structure of [22]. We remove most gait constraints but retain a fixed trotting contact schedule to facilitate sim-to-real transfer. Table 1 lists the resulting reward terms, expressions, and weights. Table 2 summarizes our notation and lists the policy observation and action space.

B Hyperparameters and Architecture for Vision Module

The vision module is structured as follows: first, we run the pretrained convolutional backbone [23] on the color image to compute a feature l_t for each pixel. For those patches that have an associated terrain property label e_t from the proprioceptive traversal, we form a tuple (l_t, e_t) . We discretize the continuous e_t into bins. Finally, we train a linear model with Softmax activation to predict the bin associated with each pixel feature. Training parameters are given in Table 4.

C Simulated Evaluation

We collect five minutes of simulated data and train a vision module on ice, gravel, brick, and grass, assigning them arbitrary friction coefficients of $\mu = \{0.25, 1.17, 2.08, 3.0\}$ respectively. Qualitatively, the vision module 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 4 shows that frictions in this range have less influence on locomotion performance. In contrast, the vision module trained on data from our Active Sensing Motor Policy learns to distinguish all four terrains. Quantitatively, ASMP results in lower dense prediction loss on images from a held-out test trajectory (Figure 10, Appendix).

D Path Planning Procedure

We use the A* algorithm [25] to compute cost-minimal paths.



Figure 9: Example hand-labeled segmentation maps used for real-world performance analysis. Orange=pavement, yellow=dirt, blue=grass, purple=gravel.

Hyperparameter	Value
discount factor	0.99
GAE parameter	0.95
# timesteps per rollout	21
# epochs per rollout	5
# minibatches per epoch	4
entropy bonus (α_2)	0.01
value loss coefficient (α_1)	1.0
clip range	0.2
reward normalization	yes
learning rate	$1e - 3$
# environments	4096
# total timesteps	2.58B
optimizer	Adam

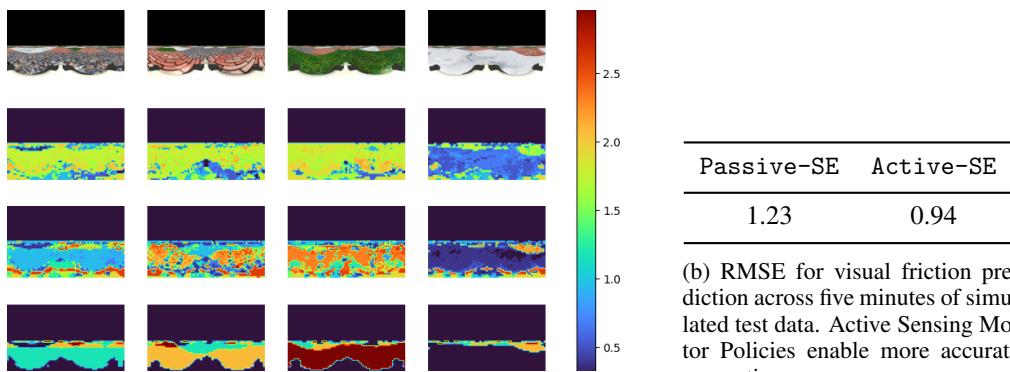
Hyperparameter	Value
framerate	5 fps
learning rate	$1e - 3$
batch size	64
# epochs	20
optimizer	Adam
layers	1
activation	Softmax
# discrete categories	20

Table 4: Vision module training hyperparameters.

Table 3: PPO hyperparameters.

Table 5: Numerical result of real-world friction prediction evaluation (Table 6).

Surface	Measured	ASMP (Ours)	Passive (Baseline)	Vision (Train)	Vision (Test)
Grass	1.45	1.92 ± 0.28	2.58 ± 0.14	1.89 ± 0.18	1.80 ± 0.26
Pavement	0.89	1.35 ± 0.29	2.42 ± 0.13	1.32 ± 0.22	1.35 ± 0.22
Dirt	0.63	0.90 ± 0.23	2.12 ± 0.26	0.90 ± 0.23	1.09 ± 0.36
Gravel	0.74	0.90 ± 0.39	1.75 ± 0.47	0.81 ± 0.26	1.12 ± 0.39



(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 10: **Friction inference from color images in simulation.** 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.