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

Abstract: To plan efficient robot locomotion, we must use the information about a terrain's physics that can be inferred from its color, texture, and semantics. To this end, we train a visual perception module that predicts terrain properties using labels from a small amount of real-world proprioceptive locomotion. To ensure label precision, we introduce *Active Sensing Motor Policies*. These policies are trained to prefer motor skills that facilitate accurately estimating the environment's physics, like swiping a foot to observe friction. The estimated labels supervise a vision model that infers physical properties directly from color images and can be reused for different tasks. Leveraging a pretrained vision backbone, we demonstrate robust generalization in image space, enabling path planning from overhead imagery despite using only ground camera images for training.

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 use a combination of proprioception and depth sensing to perceive the ground beneath the robot or obstacles around it [6, 7, 8, 9, 10, 11, 12, 13, 14] but discard a valuable signal about remote terrain's non-geometric physical properties that is conveyed by color and texture. To plan an efficient traversal route for a legged robot that optimizes e.g. its energy efficiency, safety, or maximum speed, it is necessary to predict the relevant physical properties of upcoming terrain by inferring them from color images.

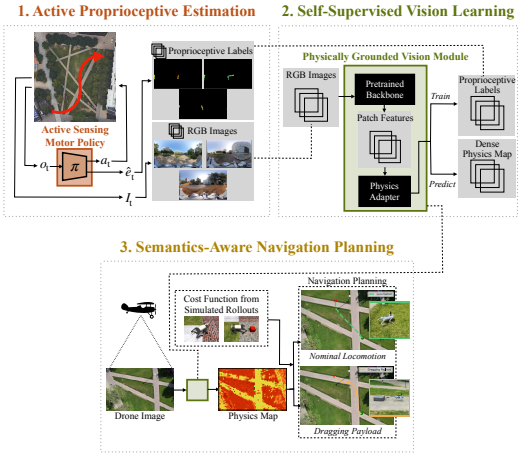


Figure 1: **Learning to see how terrains feel.** We propose (1) learning an optimized gait for collecting informative proprioceptive terrain labels that are (2) used to supervise training for a vision module, which can (3) be used for navigation planning with new tasks and image sources.

The real-world relationship between terrain properties and visual appearance in color images can vary across environments, so static or offline datasets may not capture it appropriately. Learning this relationship from a small amount of direct experience in the target environment is preferable. To make the most of the data we collect, we would like each experience to inform the traversal of the robot not only when walking normally but also when performing a new task (like payload dragging) or optimizing a new objective (like a preference for speed vs energy efficiency). Therefore, unlike prior works in semantic terrain perception that consider a traversability metric for a specific policy and task [15, 16, 17, 18], we explicitly estimate terrain physics parameters that are invariant to the policy and task (i.e. friction coefficient). A natural way of estimating labels for the terrain's physical parameters is by training a neural network to predict the parameters from the proprioceptive history, supervised by the ground truth labels available in simulation [4, 6]. We

7th Conference on Robot Learning (CoRL 2023), Atlanta, USA.

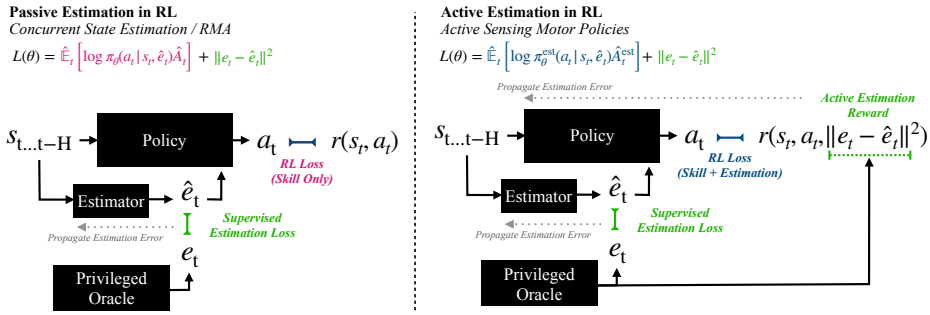


Figure 2: **Active Sensing Motor Policies optimize for estimation.** Unlike passive methods (left), which estimate the state only to the extent that it is observable as a byproduct of control relevance, Active Sensing Motor Policies (right) directly incentivize improved state estimation through the advantage function. This incentivizes a policy to adopt information-gathering behaviors, like intentionally swiping the robot's foot during legged locomotion, to improve the estimate quality.

discovered that the estimates obtained through this approach can be imprecise because the robot's behavior often renders the terrain properties unobservable. Instead, we propose training a specialized 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.

We use the improved data obtained through ASMP to train a more accurate visual perception module that can generate efficient plans for different operating states, like dragging heavy objects, by considering the impact of terrain properties on traversal cost. Recent advancements in computer vision have offered increasingly general solutions to segmentation [19, 20, 21] and feature extraction [22, 23]. Building upon this progress, our work grounds a pre-trained backbone of vision features learned from internet data in the physics governing robot performance. The resulting model predicts the physical properties of upcoming terrain, which can then be taken into account for path planning. Despite being trained solely with images captured from an onboard camera, our resulting model can also be evaluated to predict terrain properties using images from various cameras and viewpoints. This enables path planning using images from a drone, flying overhead, which can see a larger ground area ahead of the quadruped.

2 Method

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

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)

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, 6, 8] or explicit [4, 24] 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], 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 randomized dynamics or unobserved state variables [4].

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*: $r_{\text{est}} = c \cdot \exp\left(\|e - \hat{e}\|^2\right)$. 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 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 $1\,\mathrm{m}$ and $5\,\mathrm{m}$ 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. 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 $\mathbf{I}_t^e$ for each color image $\mathbf{I}$ where the traversed pixels are labeled with their measured dynamics.

For each color frame $\mathbf{I}_t$, we use the pretrained convolutional backbone [21] to compute a dense feature map. Similar to the procedure that Oquab et al. [23] 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 [24] 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. 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.

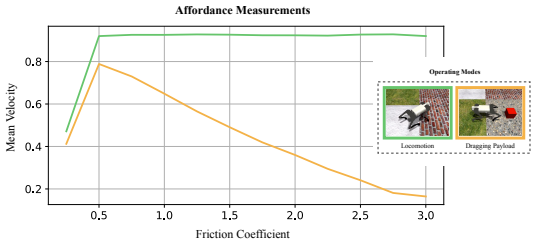


Figure 3: **Locomotion affordances.** We measure the dependence of locomotion performance ($1\,\mathrm{m/s}$) on terrain friction in two different operating modes. In free locomotion, the controller maintains the target velocity across a range of friction coefficients, except for the lowest friction. In contrast, when dragging a weighted box, the robot slows down as the terrain friction increases.

2.4 Integrated Dynamics-Aware Path Planning from Vision

Our perception module (Section 2.2) runs in real-time ($2\,\mathrm{Hz}$) 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 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 [25] to compute the minimum cost traversal path for the current operating state.

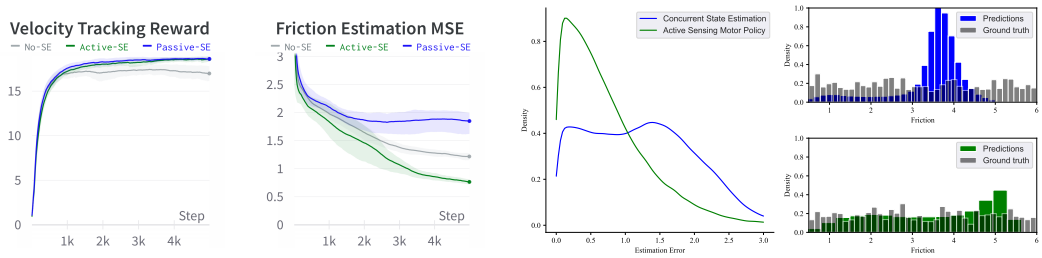
2.5 System Setup

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 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.



(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.

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 4).

Observing passive state estimates can degrade the state 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 suggest 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 a better adaptive facility is more likely to avoid slipping 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 superior policy 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 10 compares the resulting visual perception module learned from the policies performing passive vs. active estimation. 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 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 10, Appendix).

Real-world Training. We collect 15 minutes of real-world traversal data spanning diverse terrains: grass, gravel, dirt, and pavement. Following the procedure in Section 2.2, we project the traversed

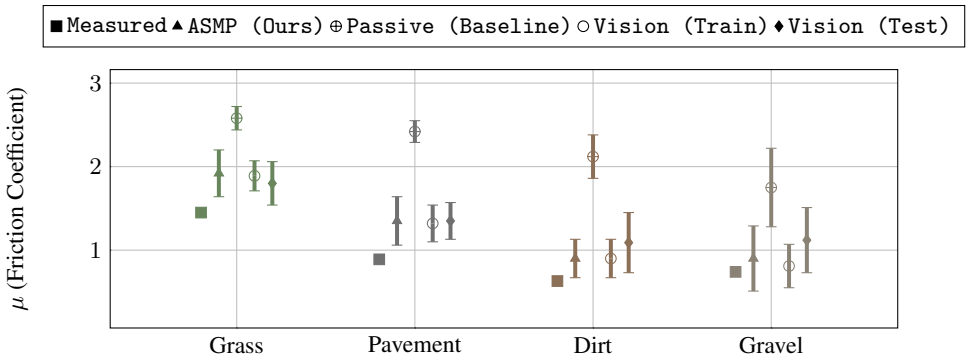


Figure 5: **Real-world friction sensing performance with proprioception and vision.** `Measured` values are directly measured by a dynamometer. The predictions from our proposed `ASMP (Ours)` agree more strongly with the dynamometer measurements than the typical policy training `Passive (Baseline)`. `Vision (Train)` reports the distribution of predictions from the vision module on the training images from the onboard camera; Evaluation on drone videos (`Vision (Test)`) incurs increased variance but preserved ordering among terrains. We use manual segmentation labels (Appendix Figure 9) to compute metrics for each terrain. Error bars indicate one standard deviation.

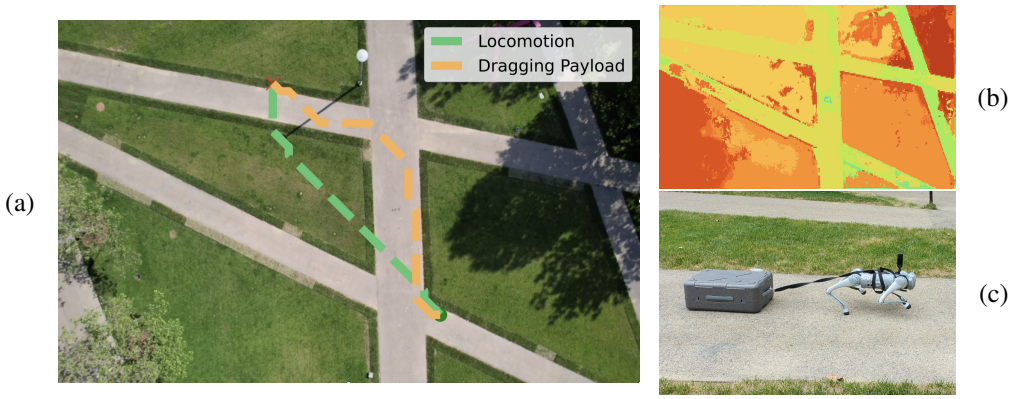
points into the corresponding camera images and train a linear head on top of a convolutional backbone trained for segmentation [21] to predict the terrain friction estimate for each traversed patch.

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 5). To obtain a ground truth friction value for comparison, we use dynamometer to measure the weight of a payload made of robot foot material and its drag force across each terrain. 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 [24] 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 $20\,\mathrm{s}$ 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. 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 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.



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.** (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 robustness of the pretrained vision model enables inference of terrain friction with a different camera model and viewing pose. (c) 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.

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 [16, 17, 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.

Some works have demonstrated self-supervised terrain segmentation from proprioceptive data [15, 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. [15] 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 in an unsupervised manner 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) thanks to our Active Sensing Motor Policies, we can directly predict the terrain properties instead of a traversability 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 an adaptation response from semantic visual information to improve low-level locomotion performance [18, 29]. Loquercio et al. [29] 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. 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. [18] trained a semantic visual perception module for legged quadrupeds using human supervision.

The resulting system adapted a scalar quantity in 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]. These works train an omni-policy for all robots and environments, enabling interesting zero-shot generalization but not explicitly adjusting for embodiment, operating state, or camera viewpoint variation.

Estimation Mode	Friction Loss	Roughness Loss	Torque Penalty
`None (Passive)`	1.00	1.00	−0.34
`Friction`	0.47	1.06	−0.87
`Roughness`	0.99	0.72	−0.84
`Joint Fr.+Ro.`	0.49	0.80	−1.18

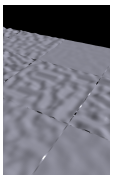


Figure 7: **ASMP for multiple physical parameters.** Estimation of friction and roughness is significantly improved by ASMP, even when both parameters are jointly targeted. We report normalized estimation loss for passive estimation (`None (Passive)`), rewarding estimation of each parameter separately (`Friction`, `Roughness`), and rewarding estimation for both parameters in a single policy (`Joint Fr.+Ro.`). Variation in torque across policies reflects that a change in motor strategy enabled the improved estimation.

Several works visually estimate the geometry or contact properties of the terrain and then compute the traversal cost from these metrics [33, 34]. The wheeled robots in these works have relatively limited controller design spaces. Consequently, the question of selecting a locomotion controller to gather the most informative label data has not been directly addressed. Active perception, in which a robot agent acts to increase environmental observability, offers a framework for understanding this problem. This approach has been applied to vision systems [35, 36], and more recently has been extended to include physical interaction [37, 38, 39]. 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 source of performance variation for practical locomotion tasks. To account for other parameters besides friction that vary in the environment, our framework can potentially be extended to include them. Figure 7 shows that ASMP can improve the estimation of a terrain roughness parameter in addition to friction.

In general, ASMP may be applied 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 have an impact on the terrain's visual appearance, it may be impossible to learn a vision module of the form we propose for those parameters. If the mapping varies over time, future work could explore representing uncertainty or performing fast online adaptation of the estimates to the current environment based on new proprioceptive information. Finally, our pipeline does not yet account for terrain geometry and occlusions. In the future, ASMP may also be useful to complement data from other sensors like LiDAR to achieve the most detailed perceptual representation that includes geometric information.

6 Conclusion

Proprioceptive self-supervision is a promising data source for robots to learn 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 estimation quality and the corresponding performance of a grounded vision module that is reusable for new sensor configurations and physical tasks.

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/). 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] Y. Ji, G. B. Margolis, and P. Agrawal. Dribblebot: Dynamic legged manipulation in the wild. *arXiv preprint arXiv:2304.01159*, 2023.

[6] 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.

[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] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] J. Xu, S. Liu, A. Vahdat, W. Byeon, X. Wang, and S. De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2955–2966, 2023.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[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] 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.

[34] 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.

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

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

[37] 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.

[38] 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.

[39] 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.

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

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

Table 2: Notation, observation and action space.

Parameter	Definition	Units	Dimension
	Learned Policies		
$\pi_{\texttt{No-SE}}$	No State Est	-	-
$\pi_{\texttt{Passive-SE}}$	Passive State Est	-	-
$\pi_{\texttt{Active-SE}}$	Active State Est	-	-
$\pi_{\texttt{Loco}}$	Policy from [24]	-	-
	Policy Observation ($\mathbf{o}$)		
$\mathbf{q}$	Joint Angles	rad	12
$\dot{\mathbf{q}}$	Joint Velocities	rad/s	12
$\mathbf{g}$	Norm Gravity, Body Frame	m/s^2	3
$\boldsymbol{\psi}_t$	Body Yaw, Global Frame	rad	1
$\mathbf{a}_{t-1}$	Previous Action	-	12
$\boldsymbol{\theta}^{\text{cmd}}$	Timing Reference [24]	-	4
$\hat{e}_t$	Estimator Output	-	5
	Policy Action ($\mathbf{a}$)		
$\mathbf{q}_{\text{des}}$	Joint Position Targets	rad	12
	Other Quantities		
$\mathbf{f}^{\text{foot}}$	Foot Vertical Force	N	1
$\mathbf{v}_{\text{xy}}^{\text{foot}}$	Foot xy-Velocity	m/s	1
$\boldsymbol{\kappa}$	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
$\boldsymbol{\omega}_z^{\text{cmd}}$	Target Yaw Velocity	rad/s	1
$\boldsymbol{\omega}_z$	Actual Yaw Velocity	rad/s	1
$\boldsymbol{\tau}$	Joint Torque	N m	1

A Reward Function for Policy Training

We follow the reward structure of [24]. 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 [21] 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 Qualitative Tests on Out-of-Distribution Images

Additional examples of friction prediction on out-of-distribution images are shown in Figure 8. The top row shows the original color image, while the bottom row overlays the colormapped pixel-wise friction coefficient estimate. All images are taken from the same courtyard space traversed during training, but on different days, different camera models, and different viewing angles. We do not have access to ground-truth friction labels for these environments, but the outputs for the first five images qualitatively match the in-distribution labels for grass, dirt, and different types of pavement. The sixth image is an apparent generalization failure of the vision module; the proprioceptive training labels for gravel were lower friction (blue) than the pavement (reddish-yellow), but the vision module's prediction in this high drone image is that the gravel has higher friction (red).

D Path Planning Procedure

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

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

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

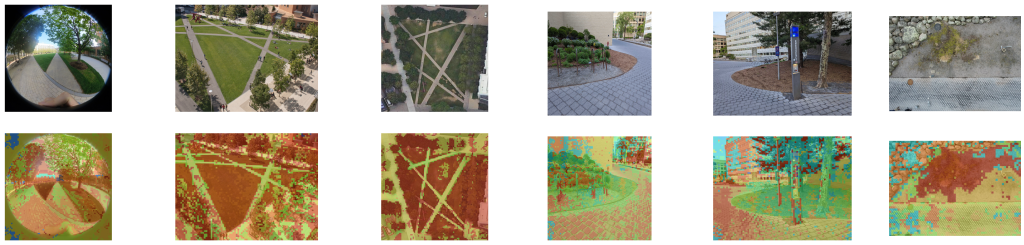


Figure 8: Example predictions on the test terrains from out-of-distribution viewing angles and camera models. From left to right: Fisheye camera image; Image from the internet; Image from google maps satellite view; Photo taken by phone (x2); Drone image.

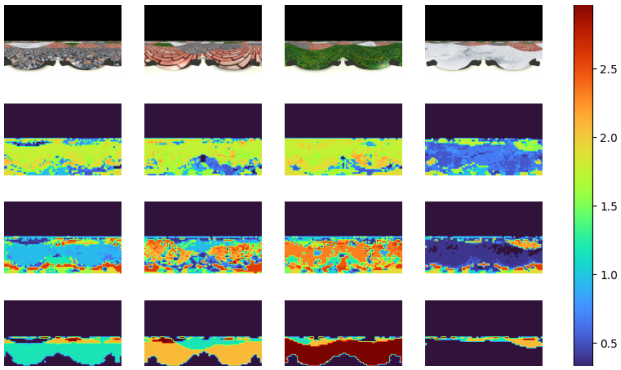
Figure 9: Example hand-labeled segmentation maps used for real-world performance analysis. Orange=pavement, yellow=dirt, blue=grass, purple=gravel.

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

Table 3: PPO hyperparameters.

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.



(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.** 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.