2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

How Does It Feel? Self-Supervised Costmap Learning for Off-Road Vehicle Traversability

Mateo Guaman Castro[1], Samuel Triest[1], Wenshan Wang[1], Jason M. Gregory[2], Felix Sanchez[3], John G. Rogers III[2], and Sebastian Scherer[1]

***Abstract*— Estimating terrain traversability in off-road environments requires reasoning about complex interaction dynamics between the robot and these terrains. However, it is challenging to create informative labels to learn a model in a supervised manner for these interactions. We propose a method that learns to predict traversability costmaps by combining exteroceptive environmental information with proprioceptive terrain interaction feedback in a self-supervised manner. Additionally, we propose a novel way of incorporating robot velocity into the costmap prediction pipeline. We validate our method in multiple short and large-scale navigation tasks on challenging off-road terrains using two different large, all-terrain robots. Our short-scale navigation results show that using our learned costmaps leads to overall smoother navigation, and provides the robot with a more fine-grained understanding of the robot-terrain interactions. Our large-scale navigation trials show that we can reduce the number of interventions by up to 57% compared to an occupancy-based navigation baseline in challenging off-road courses ranging from 400 m to 3150 m. Appendix and full experiment videos can be found in our website: https://mateoguaman.github.io/hdif.**

I. Introduction

Outdoor, unstructured environments are challenging for robots to navigate. Rough interactions with terrain can result in a number of undesirable effects, such as rider discomfort, error in state estimation, or even failure of robot components. Unfortunately, it can be challenging to predict these interactions a priori from exteroceptive information alone. Certain characteristics of the terrain, such as slope, irregularities in height, the deformability of the ground surface, and the compliance of the objects on the ground, affect the dynamics of the robot as it traverses over these features. While these terrain characteristics can be sensed by proprioceptive sensors like Inertial Measurement Units (IMUs) and wheel encoders, these modalities require direct contact with the terrain itself. Additionally, the robot's interaction with the ground leads to dynamic forces which are proportional to velocity and suspension characteristics. In order to *feel* what navigating over some terrain at some velocity is like, we argue that the robot must actually traverse over it.

Fig. 1: We present a costmap learning method for off-road navigation. We demonstrate the efficacy of our method on a large ATV (left), and on a Warthog UGV robot (right).

Previous approaches for off-road traversability have focused on representing exteroceptive information as occupancy maps [1,2], or learning semantic classifiers from labeled data to map different terrains to different costs in a costmap [3,4]. Yet, this abstracts away all the nuance of the interactions between the robot and different terrain types. Under an occupancy-based paradigm, concrete, sand, and mud would be equally traversable, whereas tall rocks, grass, and bushes would be equally *non*-traversable. In reality, specific instances of a class may have varying degrees of traversability (e.g. some bushes are traversable but not all).

Other approaches have characterized terrain roughness directly from geometric features [5–7]. Yet, what we are really interested in capturing is roughness as the vehicle experienced it, rather than capturing the appearance or geometry of roughness. For instance, a point cloud of tall grass might appear rough, but traversing over this grass could still lead to smooth navigation if the terrain under the grass is smooth. Finally, other learning-based methods learn predictive models or direct control policies for off-road navigation [8]. However, many of these do not take into account robot dynamics, which are fundamental in scenarios where the state of the robot, such as its velocity, can lead to a wide range of behaviors. For instance, speeding up before driving over a bump in the terrain can lead to jumping, whereas slowing down will usually result in smoother navigation.

In this paper, we propose a self-supervised method that predicts costmaps that reflect nuanced terrain interaction properties relevant to ground navigation. Motivated by examples in legged locomotion [9], we approach this problem by learning a mapping from rich exteroceptive information and robot velocity to a continuous traversability cost derived

*Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement number W911NF-21-2-0152. 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 Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

[1] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. {mguamanc, striest,wenshanw,basti}@andrew.cmu.edu
[2] DEVCOM Army Research Laboratory, Adelphi, MD, USA. {jason.m.gregory1.civ, john.g.rogers59.civ}@army.mil
[3]Booz Allen Hamilton, McLean, VA 22102, USA sanchez_felix@bah.com

from IMU data. We propose a learning architecture which combines a CNN backbone to process high-dimensional exteroceptive information with a feed-forward network that takes in a Fourier parameterization of the low-dimensional velocity information, inspired by recent advances in implicit representation learning [10,11].

Our main contributions are: a) learning to aggregate visual, geometric, and velocity information directly into a *continuous-valued* costmap, without human-annotated labels, and b) demonstrating ease of integration into traditional navigation stacks to improve navigation performance, due to our choice of using a top-down metric representation. In the rest of this paper, we present our contributions in more detail as follows:

- We present an IMU-derived traversability cost that can be used as a self-supervised pseudo ground-truth for training.
- We demonstrate a novel way to combine low-dimensional dynamics information with high-dimensional visual features through Fourier feature mapping [11].
- We propose a system that produces continuous-valued learned costmaps through a combination of visual and geometric information, and robot velocity.
- We validate our method on outdoor navigation tasks using two different ground robots.

II. Related Work

There exists over a decade of work in learning methods for off-road traversability [12–18], much of which can be traced to the DARPA LAGR [19] and Grand Challenge [20] programs. These methods learned lightweight terrain traversability classifiers based on visual and geometric appearance [12–14,16–18]. Other approaches directly estimate a terrain roughness score by analyzing planarity [6] or eigenvalues of the terrain point cloud [5].

Much of the recent literature on learning for off-road traversability estimation has focused on semantic segmentation of visual data into discrete classes [3,21]. However, it is not immediately obvious how to map human semantic classes into costs that can be directly used for path planning and control, which often results in hard-coded mappings. More recent work by Shaban et al. [22] aims to alleviate the need for explicit mappings from human semantic classes to costs by directly learning discrete traversability classes, such as low-cost and high-cost, in a metric space from geometric data. Yet, these approaches require a large amount of labeled semantic data for training, and lack expressiveness given the limited number of traversability classes.

Recently, other works have looked into learning predictive models, control policies, and risk-aware costmaps directly from visual and multimodal inputs for navigation in challenging off-road environments. While BADGR-based methods [8,23] learn a *boolean* predictor for whether a specific sequence of actions will lead to a bumpy trajectory, we learn a continuous value for traversability that is aggregated into a costmap that can be used directly to optimize trajectories, without the need to query the network for every sample of our MPPI optimizer [24]. Triest et al. [25] learn a neural network-based dynamics model for a large ATV vehicle and explore the use of different types of multimodal data as input to their neural network. Sivaprakasam et al. [26] learn a dynamics model in simulation to derive a dynamics-aware cost function for downstream planning tasks. Fan et al. [27] learn a traversability risk costmap from lidar data, and Cai et al. [28] learn a speed distribution map that is converted into a costmap using a conditional value at risk (CVaR) formulation. Triest et al. [29] learn CVaR-based uncertainty-aware traversability costmaps from lidar data using inverse reinforcement learning.

Most recently, traversability estimation for off-road robots has shifted towards learning continuous costmaps in a self-supervised manner with IMU signals as learning targets, with these methods learning from RGB data [9,30,31], or point clouds [32], and [30,32] conditioning on robot speed similar to our approach. We use both RGB and point cloud data in our approach to generate high-resolution, continuous costmaps, and we demonstrate our approach in large-scale, challenging off-road courses at much higher speeds on two different large robot platforms.

III. Costmap Learning

We now introduce our self-supervised costmap learning method, which associates a proprioception-derived cost to high-dimensional visual and geometric data, and robot velocity. In the following section, we will describe our costmap learning pipeline (Figure 2), which consists of:

1) Derivation of a cost function from proprioception,
2) Extraction of a map-based representation from visual and geometric data,
3) Representation of velocity as an input to our model, and
4) Training a neural network to predict traversability cost.

A. Pseudo-Ground Truth Traversability Cost

We aim to learn a continuous, normalized traversability cost that describes the interactions between the ground and the robot, and which can be directly used for path planning. As shown in previous work [9,15,18], linear acceleration in the z axis, as well as its frequency response, capture traversability properties of the environment that not only depend on the characteristics of the ground, but also on the speed of the robot. To obtain a single scalar value that generally describes traversability properties of the terrain, such as the roughness, bumpiness, and deformability, we use the bandpower of the IMU linear acceleration of the robot in the z axis:

$$\hat{c} = \int_{f_{\min}}^{f_{\max}} S_{a_z}^{W}(f)\, df, \tag{1}$$

where $\hat{c}$ is our estimated traversability cost, $S_{a_z}^{W}$ is the power spectral density (PSD) of the linear acceleration a_z in the z axis, calculated using Welch's method [33], , and $f_{\min}$ and $f_{\max}$ describe the frequency band used to compute bandpower. We use a frequency range of 1-30 Hz, since this range highly correlates with human-labeled roughness scores, and normalize based on data statistics from recorded trajectories, as detailed in the Appendix.

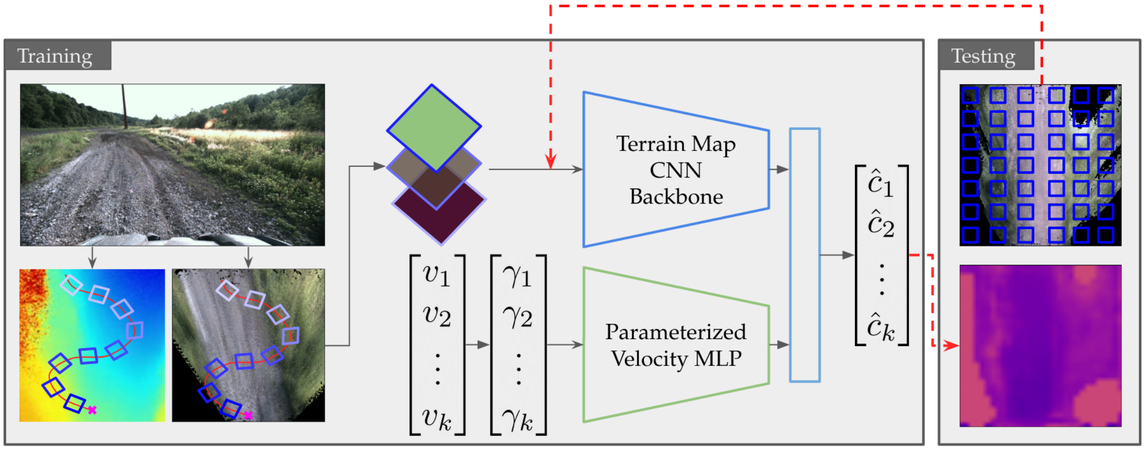

Fig. 2: **System Overview**: During training, the network takes in patches cropped from a top-down colored map and height map along the driving trajectory, as well as the parameterized velocity corresponding to each patch. The network predicts a traversability cost for each patch, supervised by a pseudo ground-truth cost generated from IMU data. During testing, the whole map is subsampled into small patches, which are fed into the network to generate a dense, continuous costmap.

B. Mapping

We represent the exteroceptive information about the environment in bird's eye view (BEV), which allows us to aggregate visual and geometric information in the same space, which we refer to as the "local map." We use a stereo matching network [34] to obtain a disparity image, from which we estimate the camera odometry using TartanVO [35], a learning-based visual odometry algorithm. We use this odometry and the RGB data to register and colorize a dense point cloud which we then project into a BEV local map. The local map consists of a stacked RGB map, containing the average RGB value of each cell, and a height map, containing the minimum, maximum, mean, and standard deviation of the height of the points in each cell, ignoring all points 2 meters above the ground surface to deal with overhangs. Additionally, we include a boolean mask that describes areas in the local map for which we have no information, either due to occlusions or limited field of view of the sensors.

C. Velocity Parameterization

At high speeds, rough terrain leads to higher shock sensed by the vehicle, proportional to its speed, as explored in [18]. We obtain velocity-conditioned costmaps by including robot velocity as an input to our network.

In order to balance the high-dimensional local map input and the low-dimensional velocity input, we use Fourier feature mapping [11] on the robot's velocity. Recent advances in implicit neural representations have shown that mapping a low-dimensional vector (or scalar) to a higher dimensional representation using Fourier features can modulate the spectral bias of MLPs (which is usually biased towards low-frequency functions [36]) towards higher frequencies by adjusting the scale of the Fourier frequencies. Intuitively, we hypothesize that this parameterization lets the netwrok learn a function that more readily adjusts to subtle changes in velocity input, and prevents the network's predictions from being dominated by the high-dimensional 2D inputs. We use the following parameterization:

$$\gamma(v) = \begin{bmatrix} \cos(2\pi b_1 v) \\ \sin(2\pi b_1 v) \\ \vdots \\ \cos(2\pi b_m v) \\ \sin(2\pi b_m v) \end{bmatrix} \tag{2}$$

In Equation 2, v corresponds to the norm of the 3D velocity vector, $b_i \sim \mathcal{N}(0, \sigma^2)$ are sampled from a Gaussian distribution with tunable scale σ, and m corresponds to the number of frequencies used to map the scalar velocity value into a $2m$ dimensional vector.

D. Costmap Learning

Our costmap learning pipeline consists of three parts: a) obtaining local map patches from robot trajectories, b) training a network to predict traversability costs from a set of patches and associated pseudo ground-truth labels, and c) populating a cost map using the trained network at test time.

Local Map Patches: We extract 2x2 meter patches (roughly the robot footprint) of the local map corresponding to the parts of the environment that the robot traversed over during the dataset generation. Since the patch under the robot is not observable from a front-facing view, we first register all the local maps into an aggregated map. We use the robot odometry to locate and extract the patch in the global map at a given 2D position and orientation. At each of these positions, we use a sliding window of the last one second of IMU linear acceleration data to obtain a pseudo ground-truth traversability cost as described in Section III-A. We also record the velocity at these positions for training.

Cost Learning: We train a deep neural network $f_\theta(P, v)$ parameterized by weights θ, as shown in Figure 2, that takes in as input local map terrain patches $P \in \mathbb{P}$ and corresponding Fourier-parameterized velocities $\gamma(v), \gamma : \mathbb{R}^+ \to \mathbb{R}^{2m}$, and predicts the traversability cost of each patch $\hat{c} =$

$f_\theta(P, \gamma), f_\theta : \mathbb{P} \times \mathbb{R}^{2m} \rightarrow \mathbb{R}^+$. We use a ResNet18 [37] backbone to extract features from the patches, and a 3-layer MLP to extract features from the parameterized velocity. Finally, we concatenate these features and pass them through a fully-connected layer with sigmoid activation to obtain a normalized scalar value representing the learned traversability cost. We train this network using a Mean Squared Error (MSE) loss between the predicted costs and the pseudo ground-truth values using the Adam [38] optimizer.

Costmap Prediction: We produce costmaps at test time by taking the current local map in front of the robot, extracting patches at uniformly sampled positions (with the same orientation as the robot's current orientation), and passing them into the network. We then reshape each of the cost predictions into a costmap that corresponds to the original local map. We find that it is important to add a stride in the sampling process to allow the patch cost querying through the network to run in real time. In our experiments, we subsample the local map with a stride of 0.2m, and upsample the reshaped predicted cost values back to the shape of the local map, which allows us to produce costmaps at 7-8 Hz on an onboard NVIDIA GeForce RTX 3080 Laptop GPU.

IV. Experimental Results

A. Training Data

To train our network, we use TartanDrive [25], a large-scale off-road dataset containing roughly 5 hours of rough terrain traversal using a commercial ATV with a sensor suite. We use the stereo images in the dataset to obtain dense point clouds using TartanVO [35] (section III-B), as well as the IMU and odometry data to obtain traversability costs and velocities, respectively. In order to effectively train our network, we find it necessary to augment and balance the data with respect to the pseudo ground-truth traversability cost. We enforce a 2:1 ratio of high to low cost frames, resulting in 15K training frames, and 3K validation frames. Additionally, we fine-tune the base model with 9.5K training frames and 1.3K validation frames collected on the Warthog platform for our Warthog experiments.

B. Navigation Stack

We validate our learned costmaps in off-road navigation tasks, where the goal is to navigate to a target location. For state estimation, we use Super Odometry [39] on the commercial ATV and a pose graph-based SLAM system on the Warthog [40]. For path planning and control, we use model predictive path integral control (MPPI) [24]. We plan through a kinematic bicycle model with actuator limits on both the ATV and the Warthog. In order to obtain costs that can be used for the MPPI optimization objective, we query the learned costmap via Equation 3. This cost function queries the costmap for each state-action pair in the trajectory τ and sums it with a weighted Euclidean distance between the final state and the goal g (where $p(s)$ extracts the x-y position of state s). Since our learned costmap only learns costs for parts of the terrain it has driven over, it will not know what cost to assign to obstacles that the robot is

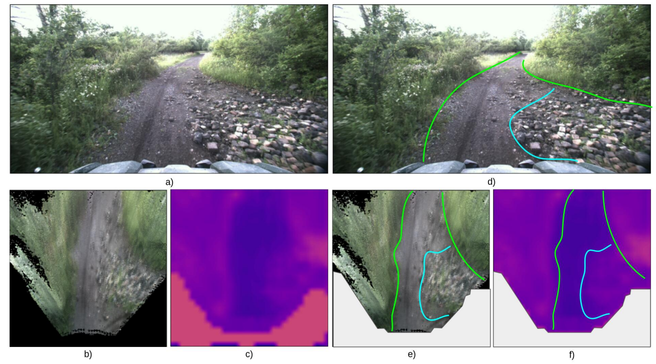

Fig. 3: Learned Traversability Costmaps: a), b), and c) show the robot's front-facing view, the corresponding RGB map, and the predicted costmap respectively. d), e), and f) show a manually annotated version of a), b), and c) for easier visualization of the different traversability properties that our costmap captures. We set the unknown regions to have a cost of 0.5 as shown in c), which we mask out in f).

incapable of traversing over. We alleviate this by composing our learned costmap with a lethal height costmap for obstacle avoidance (with a high threshold), resulting in costmap J_{map}. K_g is found empirically.

$$J(\tau) = \sum_{t=0}^{T} [J_{map}(s_t, a_t)] + K_g ||(p(s_T) - g)||_2 \qquad (3)$$

C. Robot Platforms

We demonstrate our system on two different ground robots autonomously operating at 3 m/s: a large all-terrain autonomous commercial vehicle (ATV), and a Clearpath Warthog robot [41], a large unmanned ground vehicle (UGV). For more hardware details, please see our Appendix.

D. Results

In our experiments, we are interested in answering the following questions:

1) Do our learned costmaps capture more nuance than the baseline lethal-height costmaps?
2) How much of an effect does velocity have in our predicted costmaps?
3) Do we obtain more intuitive behaviors and better navigation performance using learned costmaps inside a full navigation stack?
4) How well does our method transfer to other robots?

Do learned costmaps show more nuance? To observe trends in the learned costmap, we tele-operated the ATV around our test site and visually analyzed the predicted costmaps in different environments. We observe that our learned costmaps are able to estimate different costs for terrains of the same height but with different traversability properties. In Figure 3, we show a scenario with gravel, smooth dirt, and vegetation, where our costmap predicts different costs for these different terrains.

We notice some particular trends in our experiments. Transitions in texture from one terrain to another are usually

Roughness Estimator	Val. Loss ($\times 10^{-2}$)
σ_z [42] $+ v$	6.08
σ_z [42] $+$ RGB $+ v$	5.92
Ours	**4.82**

TABLE I: Comparison to other appearance-based roughness estimators. Our method outperforms both a geometric, as well as a geometric and visual roughness estimators.

discernible in our costmaps. Terrains covered in grass exhibit a higher cost than smooth dirt paths. Finally, terrains with higher frequency textures, such as large patches of gravel, are predicted to have higher costs than smooth terrains.

Does velocity affect the predicted costmaps? We evaluate whether adding velocity as an input to our network improves the loss achieved in the validation set. We analyze three different models:

1) **patch-model**: just the ResNet backbone for patch feature extraction.
2) **patch-vel-model**: combines the ResNet backbone with an MLP to processes normalized velocity.
3) **patch-Fourier-vel-model**: combines the ResNet backbone with an MLP that processes Fourier-parameterized normalized velocity.

Training results for all three models and ablations are shown for five random seeds in the Appendix. We find that adding velocity as an input to the network leads to better performance than using local map patches alone, and that the `patch-Fourier-vel-model` performs best as measured by the validation loss. Our ablations show that using both RGB and height statistics performs slightly better than using just RGB, and that the scale and number of frequencies used for Fourier parameter mapping of velocity do not make much of a difference.

We compare our method with two appearance-based roughness baselines to evaluate whether our network captures more accurately the robot-terrain interactions. The first baseline characterizes roughness as the standard deviation of height in a given patch [42]. We extend this metric with the average RGB values in a patch for our second baseline. For a fair comparison, we add normalized velocity as an input to both baselines. We learn the weights for a linear combination of these inputs through logistic regression. We compare with our best learned model in the validation set (Table I).

We also evaluate the effect of robot speed in the predicted costmaps in a real-robot experiment. In this experiment, we command the ATV different velocities and aggregate the average predicted cost over a straight 200 m trail with similar terrain characteristics throughout. Additionally, we integrate the sum of costs of the entire costmap (energy) along the trajectory. We summarize the results in Table II, and show the resulting costmaps in the Appendix. We find that traversability cost and overall costmap energy generally increase as robot speed increases.

Do learned costmaps improve *short-scale* navigation? We deploy our learned costmaps within a full navigation stack in two different navigation tasks, and compare the performance against a navigation stack which uses just a baseline geometric occupancy-based costmap. When using

Velocity	Pred. Traj. Cost	GT Traj. Cost	Costmap Energy
3 m/s	0.093	0.038	0.185
5 m/s	0.163	0.055	0.250
7 m/s	0.201	0.084	0.289
10 m/s	0.195	0.106	0.266

TABLE II: Effect of velocity on the predicted costmap in a real-robot experiment. Higher speeds generally result in higher costs, both on the robot's trajectory and the costmap as a whole. Note that while the IMU-based ground truth cost increases monotonically as velocity increases, the predicted cost stops increasing after 7 m/s likely due to lack of enough training data at higher speeds.

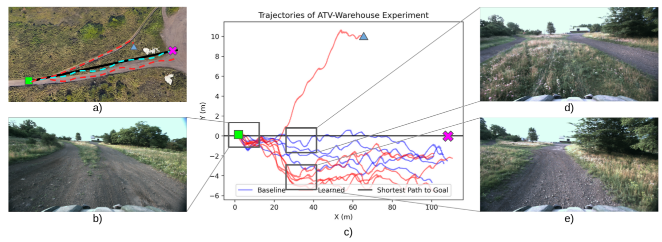

Fig. 4: Short-scale navigation experiment (ATV-Warehouse). When using our costmap, the robot deviates from the straight path to the goal and chooses the smoother dirt path (e) over the patch of vegetation (d) to get to the goal. a) Sketch of the trajectories taken when using the baseline stack (blue) and our stack (red). The green square is the start position, and the pink cross is the goal position. One trajectory ends in an intervention (triangle) due to the robot taking an equally-smooth dirt path to avoid the grass. b), d), and e) show front-facing views at those points in the trajectory. c) shows all trajectories, as well as the direct path to the goal (black).

Fig. 5: Overview of large-scale navigation experiments. The leftmost figure shows a satellite view of the three courses: red, blue, and green. The top row shows sample aerial views of the courses, while the bottom row shows sample first-person robot images. The color of the rectangle around the picture denotes the course.

our learned costmaps, we compose them with the occupancy-based costmap to provide basic obstacle avoidance capabilities. We set the lethal height to be 1.5 m and all unknown regions to have a cost of 0.5 for all experiments.

We set up two navigation courses for the ATV, ATV-Warehouse (Figure 4) and ATV-Turnpike, where the task is to simply move 400m and 200m straight ahead, respectively. We design these controlled courses such that a straight path would lead the robot to go over rougher terrain or patches of vegetation, but reasoning about the terrain would lead the robot to take a detour to stay on smoother trails. For ATV-Warehouse, we run five trials with our learned costmap and

Course	Nav. Stack	Avg. CTE (m)
ATV-Warehouse	Baseline	1.29 ± 1.02
ATV-Warehouse	HDIF (Ours)	3.39 ± 0.94
ATV-Pole	Baseline	2.44 ± 2.25
ATV-Pole	HDIF (Ours)	2.68 ± 0.41

TABLE III: Avg. Cross-track error (CTE) between planned paths and the shortest path to goal. Learned costmaps resulted in trajectories that deviated more from the shortest path to avoid areas of higher cost, e.g. grass or gravel.

five trials with the baseline costmap, and for ATV-Turnpike, we run three trials for each costmap.

The baseline costmap leads the robot to navigate in a straight line (with some noise), which leads straight into the patch of vegetation. Additionally, this baseline stack leads to less consistent navigation, which occurs because most of the terrain in front of the robot appears to be "traversable," since there are no obstacles above the 1.5 m lethal height, which leads to many different "optimal" paths. On the other hand, our learned costmaps cause the robot to consistently navigate around the patch, staying on marked paths, and even moving away from parts of the path with gravel towards smoother sections of the trail.

We measure the cross-track error (CTE) as a way of measuring the deviation of the chosen path from a naïve, straight path to the goal. In this case, lower is *not* necessarily better, since the experiment is explicitly designed for the robot to deviate from the straight path to find the smoother trails if it is able to reason about different terrains. We show the preferred paths in Figure 4, and numerical results in Table III. ATV-Turnpike experiment details are in the Appendix.

Do learned costmaps improve large-scale navigation? To provide a direct measurement of navigation performance, we set up three large scale navigation experiments in three challenging off-road courses, with waypoints every 50 m, and measure our navigation performance using number of interventions, as is common practice [6,43]. The safety driver was directed to only intervene when either: a) the robot missed a waypoint by over 4 m, or b) a collision with a non-traversable obstacle was imminent. These courses (Figure 5) include flat and hilly terrain, loose material ranging from fine gravel to large cobbles, and vegetation of various dimensions. For both navigation stacks, we set the target speed to 3.5 m/s, lethal height to 0.5 m and fill unknown values of our costmap to 0.5. We observe that in all three courses, our navigation stack with learned costmaps outperforms the baseline stack in terms of number of interventions, as detailed in Table IV. Our learned costmaps lead the robot to stay on smoother paths and avoid vegetation if possible, which leads to an overall decrease in intervention events. In practice, this leads to the robot taking wider turns, staying on clear paths, and avoiding rough features that lead to bumpy navigation, such as large cobbles.

How well does our method transfer to other robots? We set up two additional short-scale navigation experiments using the Warthog, with a fine-tuned model as detailed in Section IV-A. We used a separate geometry-based autonomy stack [44] as the baseline, and for our stack we simply

Navigation Stack	Course	Interventions	Course Length (m)
Baseline	Red	7	400
HDIF (Ours)	Red	**3**	400
Baseline	Blue	9	3150
HDIF (Ours)	Blue	**6**	3150
Baseline	Green	11	950
HDIF (Ours)	Green	**7**	950

TABLE IV: Number of interventions (lower is better) in three large-scale navigation courses with varying degrees of difficulty and different types of environment features, such as slopes, vegetation, and tight turns. Our How Does It Feel (HDIF) navigation stack achieves lower number of interventions in all courses compared to the baseline stack.

composed their lidar-based binary costmap with our learned costmap. Despite the Warthog being a skid-steer robot, we treat it as if it had an Ackermann steering geometry, similar to the ATV, since its cameras only face forward.

In the first experiment, the goal is to drive 50 m to a goal located straight ahead, where the straight path goes through grass, but there is a smooth concrete path available surrounding this patch of vegetation. Similar to what we observe in the ATV experiments, our costmaps allow the Warthog to reason about the traversability properties of the smooth concrete and the vegetation, and the robot chooses to take the smooth path around the grass to reach the goal.

In the second experiment, we artificially create two different paths at a fork in a forest trail. We litter the path on the left with small logs, fallen leaves, and small rocks, while we clear the path on the right of all obstacles. We observe that with the baseline stack, the robot uniformly chooses either of the paths to get to a goal at the other end of the fork in the trail. On the other hand, with our costmaps, the robot consistently chooses the clear path on the right to get to the other side of the trail. Note that a visual semantic classifier would label both of these trails as belonging to the same class, but the subtle differences in the features of the trails cause different amounts of roughness.

In both experiments, it is clear that our costmaps lead the robot to choose smoother paths by reasoning about the different traversability properties of different terrains. We urge the reader to visit our website for experiment videos, and our Appendix for more experiment details.

V. Conclusion and Future Work

In this work we present a costmap prediction system that predicts what the interactions between the ground and the robot feel like based on environmental characteristics and robot dynamics. To achieve this, we train a network that combines exteroception and information about the robot's velocity to predict a traversability cost derived directly from robot proprioception. We demonstrate that our costmaps enable for more nuanced and diverse navigation behaviors compared to a common baseline. Future work includes online adaptation for better generalization, uncertainty estimation, as well as improved representations for exteroceptive data to overcome our current perception limitations.

References

[1] Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, and D. Pathak, "Coupling vision and proprioception for navigation of legged robots," *arXiv preprint arXiv:2112.02094*, 2021.

[2] D. D. Fan, K. Otsu, Y. Kubo, A. Dixit, J. Burdick, and A.-A. Agha-Mohammadi, "Step: Stochastic traversability evaluation and planning for risk-aware off-road navigation," *arXiv preprint arXiv:2103.02828*, 2021.

[3] D. Maturana, P.-W. Chou, M. Uenoyama, and S. Scherer, "Real-time semantic mapping for autonomous off-road navigation," in *Field and Service Robotics*. Springer, 2018, pp. 335–350.

[4] D.-K. Kim, D. Maturana, M. Uenoyama, and S. Scherer, "Season-invariant semantic segmentation with a deep multimodal network," in *Field and service robotics*. Springer, 2018, pp. 255–270.

[5] J.-F. Lalonde, N. Vandapel, D. F. Huber, and M. Hebert, "Natural terrain classification using three-dimensional ladar data for ground robot mobility," *Journal of field robotics*, vol. 23, no. 10, pp. 839–861, 2006.

[6] P. Krüsi, P. Furgale, M. Bosse, and R. Siegwart, "Driving on point clouds: Motion planning, trajectory optimization, and terrain assessment in generic nonplanar environments," *Journal of Field Robotics*, vol. 34, no. 5, pp. 940–984, 2017.

[7] D. B. Gennery, "Traversability analysis and path planning for a planetary rover," *Autonomous Robots*, vol. 6, no. 2, pp. 131–146, 1999.

[8] G. Kahn, P. Abbeel, and S. Levine, "Badgr: An autonomous self-supervised learning-based navigation system," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1312–1319, 2021.

[9] 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*, vol. 4, no. 2, pp. 1509–1516, 2019.

[10] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European conference on computer vision*. Springer, 2020, pp. 405–421.

[11] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547, 2020.

[12] A. Howard, M. Turmon, L. Matthies, B. Tang, A. Angelova, and E. Mjolsness, "Towards learned traversability for robot navigation: From underfoot to the far field," *Journal of Field Robotics*, vol. 23, no. 11-12, pp. 1005–1017, 2006.

[13] A. Angelova, L. Matthies, D. Helmick, and P. Perona, "Learning and prediction of slip from visual information," *Journal of Field Robotics*, vol. 24, no. 3, pp. 205–231, 2007.

[14] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. Muller, and Y. LeCun, "Learning long-range vision for autonomous off-road driving," *Journal of Field Robotics*, vol. 26, no. 2, pp. 120–144, 2009.

[15] E. M. Dupont, C. A. Moore, E. G. Collins, and E. Coyle, "Frequency response method for terrain classification in autonomous ground vehicles," *Autonomous Robots*, vol. 24, no. 4, pp. 337–347, 2008.

[16] K. Konolige, M. Agrawal, M. R. Blas, R. C. Bolles, B. Gerkey, J. Sola, and A. Sundaresan, "Mapping, navigation, and learning for off-road traversal," *Journal of Field Robotics*, vol. 26, no. 1, pp. 88–113, 2009.

[17] M. Bajracharya, A. Howard, L. H. Matthies, B. Tang, and M. Turmon, "Autonomous off-road navigation with end-to-end learning for the LAGR program," *Journal of Field Robotics*, vol. 26, no. 1, pp. 3–25, 2009.

[18] D. Stavens and S. Thrun, "A self-supervised terrain roughness estimator for off-road autonomous driving," in *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, 2006, pp. 469–476.

[19] L. D. Jackel, E. Krotkov, M. Perschbacher, J. Pippine, and C. Sullivan, "The darpa lagr program: Goals, challenges, methodology, and phase i results," *Journal of Field robotics*, vol. 23, no. 11-12, pp. 945–973, 2006.

[20] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[21] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[22] A. Shaban, X. Meng, J. Lee, B. Boots, and D. Fox, "Semantic terrain classification for off-road autonomous driving," in *Conference on Robot Learning*. PMLR, 2022, pp. 619–629.

[23] D. Shah, B. Eysenbach, N. Rhinehart, and S. Levine, "Rapid exploration for open-world navigation with latent goal models," in *Conference on Robot Learning*. PMLR, 2022, pp. 674–684.

[24] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1714–1721.

[25] S. Triest, M. Sivaprakasam, S. J. Wang, W. Wang, A. M. Johnson, and S. Scherer, "Tartandrive: A large-scale dataset for learning off-road dynamics models," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2546–2552.

[26] M. Sivaprakasam, S. Triest, W. Wang, P. Yin, and S. Scherer, "Improving off-road planning techniques with learned costs from physical interactions," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4844–4850.

[27] D. D. Fan, A.-A. Agha-Mohammadi, and E. A. Theodorou, "Learning risk-aware costmaps for traversability in challenging environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 279–286, 2021.

[28] X. Cai, M. Everett, J. Fink, and J. P. How, "Risk-aware off-road navigation via a learned speed distribution map," *arXiv preprint arXiv:2203.13429*, 2022.

[29] S. Triest, M. G. Castro, P. Maheshwari, M. Sivaprakasam, W. Wang, and S. Scherer, "Learning risk-aware costmaps via inverse reinforcement learning for off-road navigation," *arXiv preprint arXiv:2302.00134*, 2023.

[30] 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)*. IEEE, 2022, pp. 7197–7204.

[31] X. Yao, J. Zhang, and J. Oh, "Rca: Ride comfort-aware visual navigation via self-supervised learning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7847–7852.

[32] G. G. Waibel, T. Löw, M. Nass, D. Howard, T. Bandyopadhyay, and P. V. K. Borges, "How rough is the path? terrain traversability estimation for local and global path planning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 16 462–16 473, 2022.

[33] P. Welch, "The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms," *IEEE Transactions on audio and electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.

[34] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5410–5418.

[35] W. Wang, Y. Hu, and S. Scherer, "Tartanvo: A generalizable learning-based vo," *arXiv preprint arXiv:2011.00359*, 2020.

[36] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5301–5310.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[39] S. Zhao, H. Zhang, P. Wang, L. Nogueira, and S. Scherer, "Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8729–8736.

[40] J. Gregory, J. Fink, E. Stump, J. Twigg, J. Rogers, D. Baran, N. Fung, and S. Young, "Application of multi-robot systems to disaster-relief

scenarios with limited communication," in *Field and Service Robotics*. Springer, 2016, pp. 639–653.

[41] "Warthog unmanned ground vehicle robot - clearpath," Jan 2021. [Online]. Available: https://clearpathrobotics.com/warthog-unmanned-ground-vehicle-robot/

[42] P. Fankhauser, M. Bjelonic, C. D. Bellicoso, T. Miki, and M. Hutter, "Robust rough-terrain locomotion with a quadrupedal robot," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5761–5768.

[43] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[44] "Arl autonomy stack," 2022, accessed 13-September-2022. [Online]. Available: https://www.arl.army.mil/business/collaborative-alliances/current-cras/sara-cra/sara-overview/