TravSUITE: Traversability via Self-Supervised, Uncertainty-Aware IRL and Terrain Estimation

Samuel Triest*, Amirreza Shaban†, David D. Fan†, Wenshan Wang* and Sebastian Scherer*
*Robotics Institute, Carnegie Mellon University
†Field AI
Email: striest@andrew.cmu.edu

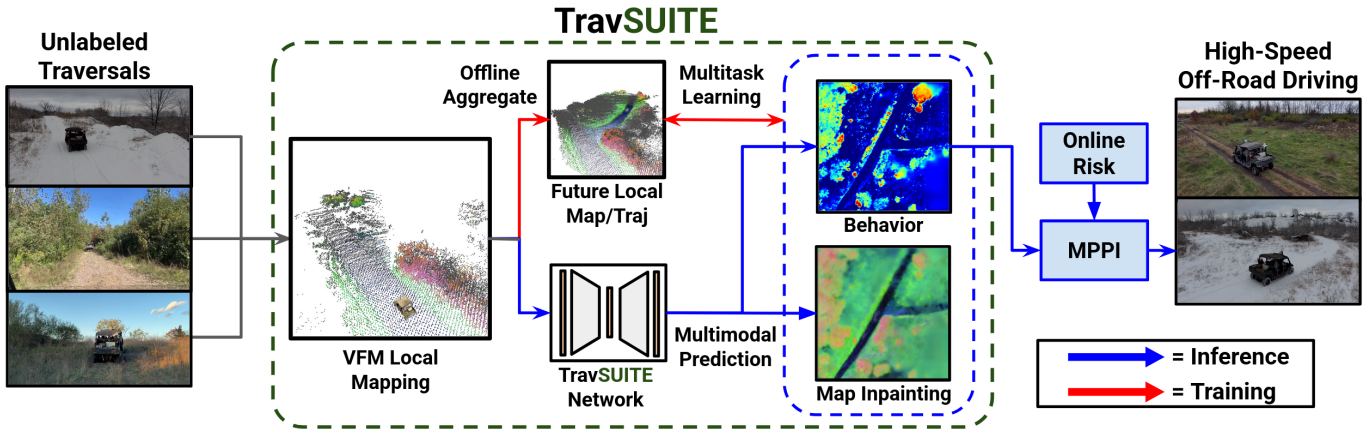

Fig. 1. TravSUITE is a learning-based off-road traversability system that expands upon prior work in off-road driving by combining off-road perception and planning tasks such as map inpainting [53, 38, 60, 70, 31] and traversability learning [47, 25, 67, 11] into a unified, self-supervised framework without any human annotation.

***Abstract*—Traversability analysis in off-road settings remains a fundamental challenge for mobile robots. Key difficulties include constructing an accurate and expressive local map from multi-modal sensor data, and using the map to design traversability rules that yield desirable navigation behavior. Importantly, this system must be resilient to the limited sensing regime brought about by complex environments and high speeds. In this paper, we present TravSUITE, a traversability system suitable for high-speed navigation in off-road environments. TravSUITE consists of two major components: 1) an efficient voxel mapper that leverages visual foundation models (VFMs) to build a rich geometric-semantic local map from streams of on-board sensor data, and 2) a unified neural network that jointly predicts traversability-relevant quantities in bird's eye view (BEV), including geometry, semantics, speed and cost. Our training strategy is entirely annotation-free and self-supervised, leveraging tasks such as map inpainting and inverse reinforcement learning (IRL) to learn both map representations and traversability. We perform a thorough ablation study and comparison to state-of-the-art approaches, and the results indicate that cost learning and auxiliary inpainting each contribute significantly to planning quality, and their combination is critical for achieving state-of-the-art performance in off-road path planning. We also design a simple risk adaptation mechanism to leverage our method's uncertainty estimates at deploy-time, and demonstrate that a combination of inpainting and risk estimation can result in 80% fewer navigation errors and 5% faster autonomous traversal speeds in real-world hardware experiments.**

I. Introduction

Many important application domains such as construction [2, 7], disaster response [50, 58, 3], forestry [10, 8], and defense [62, 5, 69], require robots to be able to navigate in off-road environments reliably and autonomously. Most practical approaches see a mobile robot 1) use its on-board sensors to build a local map of its surroundings, 2) perform traversability analysis on the local map to determine where it can and cannot traverse, and 3) use the results of its traversability analysis to plan safe paths to a desired goal. This approach is preferred in safety-critical applications due to its modularity, interpretability, and ease of incorporating additional safety constraints, and has thus been the dominant framework for performing navigation across legged [23, 25, 58, 9] and wheeled platforms [57, 5, 38, 36], for both learning [14, 26, 59, 70, 67, 58] and non-learning based [17, 23, 15, 34] methods.

While this approach may seem simple at first glance, both the local mapping and traversability problems become very challenging in complex environments such as off-road settings. The quality of the robot's local map can be significantly degraded due to factors such as observation noise and occlusion. Cost function design is a complex function of robot capability, operator preference and the environment's geometry and semantics. Effective robot navigation systems must be able to produce reliable and nuanced traversability estimates even in complex environments where the presence of observational

uncertainty may render the local map erroneous or incomplete.

There exist several recent works that aim to improve the resilience of off-road mapping pipelines to observational noise and uncertainty [53, 38, 60, 26]. These approaches typically rely on using a large neural network to predict some form of ground-truth map representation (derived from privileged data such as hindsight, hand-labeling, simulation, etc.) from the current observations. These approaches have been demonstrated to produce reliable local maps that are robust to occlusions and measurement sparsity at high speeds and long ranges [38, 26]. However, these methods often rely on robot-specific traversability data [26], or environment-specific semantics and hand-annotations [53, 38], making it challenging to generalize these approaches to novel robots or environments.

To address the challenge and tedium of hand-designing traversability rules, there has been significant prior work in learning cost functions directly from robot traversal data [5, 25, 59, 14, 32, 61, 70]. While these approaches have demonstrated promising results across many environments and robot morphologies, there lacks a common consensus on the choice of input representation (e.g. geometric, semantic, visual foundation model (VFM) features, etc.) [14, 32, 25, 61], and its effect on the downstream traversability are not well-explored. Additionally, these approaches may not account for perceptual uncertainty [47, 14, 32], and often lack large-scale datasets and dense supervision signals [25, 60, 55, 14].

TravSUITE (Fig. 1) synthesizes insights from both families of work to produce reliable and expressive traversability estimates in the presence of observational uncertainty. TravSUITE consists of three major components, 1) a flexible and efficient mapping module that produces real-time geometric-semantic map representations, 2) a framework for generating large, self-supervised, annotation-free perception and behavior datasets from robot experience, and 3) a unified neural network trained to jointly predict inpainted map representations and traversability estimates from this dataset with uncertainty. We leverage the flexibility of our mapping module and learning framework to compare against state-of-the-art approaches [38, 61, 70] and systematically evaluate what design decisions (such as mapping representation and training tasks) are important for improving path planning performance. We find that across all of our experiments, the existence of an inpainting task significantly improves traversability estimation performance, regardless of mapping representation. We also demonstrate that learning distributions of cost is important to enabling safer traversal at deploy time.

II. Related Work

The vast majority of fieldable off-road systems make use of some local map of the robot's immediate surroundings [57, 3, 38, 58, 50]. Typically, this local map is typically represented in one of two ways, as bird's eye view (BEV/2.5D), or volumetric (3D). BEV-based approaches elect to partition the robot's surroundings into "cells" of finite horizontal (xy) extent and either infinite (or hand-designed) vertical extent. Volumetric approaches partition the local environment into cells of finite xyz extent (i.e. voxels) [24, 42, 4]. In either case, cells store navigation-relevant quantities derived from sensor data, such as the elevation of the support surface [22, 21], semantic probabilities [36, 38], appearance data [14, 32] or deep features [19, 61, 31]. While BEV-based approaches sacrifice representation fidelity (e.g. overhangs, multiple support surfaces), they are often more computationally efficient and closer represent the manifold that terrestrial robots must reside on. Volumetric approaches, on the other hand, more closely represent the local environment. However, they typically are more challenging to maintain and often require additional support surface estimation techniques for practical use by terrestrial robots [34].

Similarly, various methods for encoding semantics have been proposed in the literature. The dominant approach for off-road driving is to perform semantic segmentation over a fixed set of traversability-relevant classes (e.g. trails, rocks, etc.) [37, 29, 38, 40], or traversability classes themselves [27, 53]. There has also been recent interest in encoding semantic information via VFMs such as Segment Anything [33] or Dino [41]. These approaches have the potential benefit of additional semantic granularity [61], or open-set segmentation capabilities [63]. In either case, both semantic probabilities and VFM features can be aggregated in both BEV and voxel-based approaches [37, 42, 19, 53, 38, 61, 70, 31].

There also exists a significant body of work on filling in occlusions in the local map. Early approaches rely on leveraging classical techniques such as Markov random fields [36, 65] to inpaint the terrain surface. Advances in computer vision have enabled large neural networks to learn BEV-space inpainting on a number of quantities including geometry [60, 31], semantics [53, 38], costs [26, 43] or VFM features [70]. These approaches typically rely on predicting the map representation at a fixed number of steps in the future. To the authors' knowledge, inpainting is always performed in BEV space and no work currently exists that directly inpaints volumetric representations for off-road environments.

There also exist many approaches to performing off-road traversability. A popular and straightforward approach is to hand design a cost function that relies on geometric properties of the local environment [23, 15, 34, 21, 60] such as the local slope and roughness of the terrain surface. While such an approach has been demonstrated to be effective for certain use cases and austere environments [57, 39, 44, 3, 50], a well-known failure case of these methods is false positives due to tall but compliant obstacles such as vegetation. Thus, approaches that hand-craft a traversability function based on geometry and semantics have been used to navigate in more challenging environments such as forests and snow [37, 53, 38, 26]. Another popular family of approaches relies on regressing some sort of traversability signal to the perception representation using deep learning. Many possible quantities have been proposed and demonstrated as a proxy for traversability, including experienced slip or roughness [64, 14, 25, 12, 32], simulated rollouts [24], and probability of expert demonstrations [47, 71, 59, 70, 30]. Oftentimes, it

is valuable to produce a *distribution* of costs to account for uncertainty in the perception representation or traversability function. Uncertainty in the perception can be expressed as a distribution over geometric features [21] or semantic classes [38, 53]. This uncertainty is propagated through, or incorporated in, simpler traversability rules [21, 38] to produce a cost distribution. Similarly, uncertainty in the cost function evaluation can also be expressed via reconstruction error [56, 61, 51], ensembling [47, 59], or evidential learning [12]. In all cases, this results in a *distribution* of costs per state that must be compressed into a single value for downstream path planning. It is common practice to use either expected value [47, 38], or conditional value-at-risk (CVaR) [48] to capture long-tail, worst-case costs [21, 12, 59].

TravSUITE represents an advance in state-of-the-art for autonomous local navigation in that our framework is flexible with respect to choices of both input representation and semantics. We also design our system to jointly perform map inpainting and traversability learning without any human annotations while maintaining an environment-generic representation. This enables us to perform a systematic evaluation of many design decisions, including input representation, traversability function structure, and use of inpainting tasks. Lastly, our choice of network architecture allows us to produce uncertainty estimates for test-time deployment, and is extensible to other off-road perception tasks, such as roughness/slip prediction [14, 32, 25], which we do not focus on in this work.

III. Method

A. Problem Setup

We will consider a robot tasked with navigating to a distant goal point x_G. At every discrete timestep t, the robot will have access to a state estimate x_t and observation O_t. We will assume we have access to a dataset of expert traversals that are assumed to be locally optimal. A datapoint in this dataset thus consists of three components $[O_{0:t}, x_G, \tau^E]$, which are: the set of observations $O_{0:t}$ that the robot has accrued to this point, a desired goal state x_G, and the trajectory that the expert took to reach the goal τ^E. The traversability estimation problem can be thought of as finding a cost function $f_\theta(\tau|O_{0:t}, x_G)$ under which the cost-minimizing trajectory $\hat{\tau}$ closely matches τ_E under some objective function $\mathcal{L}$ (Eq. 1).

$$\theta^* = \arg\min_\theta \mathop{\mathbb{E}}_{O_{0:t}, x_G, \tau^E \sim \mathcal{D}} [\mathcal{L}(\tau^E, \hat{\tau})] \\ s.t. \quad \hat{\tau} = \arg\min_\tau f_\theta(\tau|O_{0:t}, x_G) \quad (1)$$

In the scope of this work, we will assume that the cost function $f_\theta(\tau)$ is a weighted combination of distance to the goal x_G, and cost incurred from a "behavior map" Π_t. For the purposes of this paper, we will define our behavior map to be in BEV, and contain a cost and maximum speed for every cell (i.e. $\Pi_t = [C_t, S_t], \Pi_t \in \mathbb{R}^{2\times H\times W}$). By a slight abuse of notation, let $\Pi_t(x)$ represent a function that extracts the relevant values at state x from a map. Given this, we can perform planning by scoring trajectories, conditioned on Π_t via Equation 2 ($K_1 \approx \infty$ is used to ensure no state in the trajectory exceeds its corresponding speed limit, and K_2 is a coefficient used to balance the tradeoff between minimizing costmap cost and final-state distance to goal).

$$f_\theta(\tau|O_{0:t}, x_G) = \sum_{x\in\tau} \Big[C_t(x) + K_1 \mathbb{1}[||x_v||_2 > S_t(x)] \Big] + \\ K_2||x_T - x_G||, \quad K_1 \approx \infty \quad (2)$$

B. Decomposing Mapping and Traversability

Given the form of the cost function in Eq. 2, the goal of the traversability estimation problem is to find the best form and parameters of f_θ (and thus Π_t) that yield trajectories $\hat{\tau}$ that minimize the objective $\mathcal{L}(\hat{\tau}, \tau^E)$. It is helpful to decompose the traversability function f_θ into two components, a mapping function $M_t = h_\psi(O_{0:t})$ that compresses sensor observations into a map M_t, and a traversability function $g_\phi(M_t)$ that converts the map into the behavior map Π_t (Eq. 3).

$$\Pi_t = [C_t, S_t] = (g_\phi \circ h_\psi)(O_{0:t}), \quad \theta = [\phi, \psi] \quad (3)$$

It is important to note that the behavior map Π_t is produced via a composition of the traversability function g_ϕ and mapping function h_ψ, which each have their own parameter vector. This is important as many prior works rely on directly improving the mapping function h_ψ through auxiliary tasks such as inpainting [38, 60, 70]. In this context, these approaches can be thought of as introducing an auxiliary objective $\mathcal{L}_{aux}$ and dataset $\mathcal{D}_{aux}$ to add additional structure to $M_t = h_\psi(O_{0:t})$. Since the trajectory objective in Eq. 1 depends on the mapping representation M via Eqs. 3 and 2, it stands to reason that this auxiliary task may be able to improve the downstream traversability estimation. Ultimately, this results in a unified objective (Eq 4) in which both the trajectory $\hat{\tau}$ and map M_t are both being optimized.

$$\phi^*, \psi^* = \arg\min_{\phi,\psi} \Big[\lambda_1 \mathop{\mathbb{E}}_{O_{0:t}, x_G, \tau^E \sim \mathcal{D}} [\mathcal{L}(\tau^E, \hat{\tau}_{\phi,\psi})] + \\ \lambda_2 \mathop{\mathbb{E}}_{O_{0:t}, M^* \sim \mathcal{D}_{aux}} [\mathcal{L}_{aux}(M^*, h_\psi(O_{0:t}))] \Big] \quad (4)$$

Many state-of-the-art traversability approaches can be viewed under this framework [38, 53, 17, 70, 59, 32] through different forms of g, h and whether to supervise M, τ. An overview of our method's supervision can be found in Figure 2. In the following sections, we will describe our choice of supervision for the behavior learning and map learning components of our method.

C. Optimizing Behavior

In order to perform the behavior learning component of Eq. 4 (and to generate the behavior labels in Fig 2), we elect to use the MaxEntIRL [73] framework, in which the likelihood of a trajectory is inversely proportional to the exponent of its

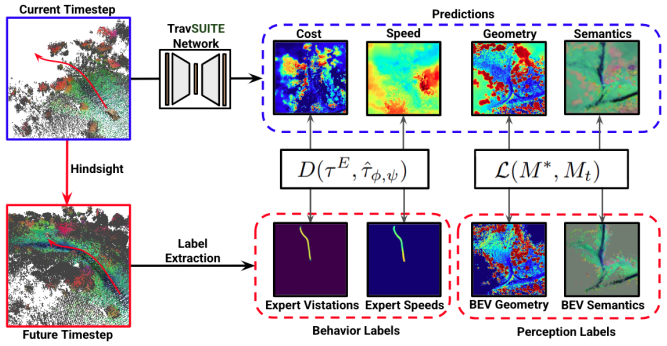

Fig. 2. An overview of our supervision. TravSUITE relies entirely on supervision from hindsight, including trajectory and map information. (Blue=derived from current observations, Red=derived from privileged information)

cost, and the objective is to maximize the likelihood of τ_E (Eq. 5). We follow a LEARCH-like procedure [47], where we sample an optimal trajectory $\hat{\tau}$ under the current costmap and goal, and then compute the gradient of the IRL objective w.r.t. the costmap to be the difference in state visitations between the expert and learner trajectories (SV is a state visitation extraction function defined in the appendix). This gradient can be used to update g_ϕ and h_ψ.

$$\mathcal{L}(\tau^E, \hat{\tau}) \triangleq p(\hat{\tau}) - p(\tau^E), \quad p(\tau) \propto exp[f_\theta(\tau|O_{0:t}, x_G)]$$
$$\frac{\partial}{\partial C} D \propto SV(\tau^E) - SV(\hat{\tau}) \quad (5)$$

We also use the expert trajectory τ_E to learn a speedmap S_t, which contains the maximum safe traversal speed for a given cell. It is important to note that we must produce an upper bound, making it necessary to predict distributions of speed rather than just the mean value. As such, we follow the example used by Cai et al. [12] by parameterizing our speedmap S_t as a categorical distribution over fixed bins, per cell. We train our model to minimize squared earth mover's distance to a delta distribution centered on the expert speed for every cell that the expert traversed. In order to encourage safe behavior, we also add a small amount of negative samples (where the target is 0) for untraversed cells (Eq. 6).

$$\mathcal{L}_S = \sum_{x \in \tau^E} [EMD^2(I(x, S_t)), \delta(||x_v||_2))] +$$
$$\lambda_s \sum_{x \notin \tau^E} [EMD^2(I(x, S_t)), \delta(0))] \quad (6)$$

In practice, we obtain labels for this behavior learning by extracting fixed-length trajectory segments from our dataset of expert traversals. For each datapoint, we sample a trajectory segment $50m$ in length and assign x_G to be the final state in the trajectory. This approach is preferred to time-based sampling as it ensures that demonstrations in more challenging scenarios (with slower expert speeds) do not have reduced path complexity. In order to sample learner trajectories $\hat{\tau}$, we use MPPI [68] under a kinematic bicycle model and the cost function from Eq. 2. Note that for training stability, we set the speed penalty (K_1) to 0 at train time.

D. Optimizing the Perception Representation

In order to instantiate the perception learning component of Eq. 4 (and the perception labels in Fig 2), we must define M_t^*. Similar to prior work [38, 70], we use hindsight supervision in order to generate labels M_t^*, and train our mapping network $M_t = h_\psi(O_{0:t})$ to predict M_t^* (Eq. 7). Note that this objective encourages inpainting as M_t^* is a function of both past and future timesteps (e.g. $M_t^* \sim O_{0:T}$).

$$\mathcal{L}_{aux} = ||M^* - M_t||, \quad M^* \sim O_{0:T}, \quad M_t \triangleq h_\psi(O_{0:t}) \quad (7)$$

IV. Implementation Details

A. Aggregating Observations with VFM Voxel Mapping

In practice, it is not feasible to maintain and process an ever-growing list of observations $O_{0:t}$ when t can span thousands of timesteps or more. To address this, we developed a voxel mapper that aggregates visual and geometric features in a fixed volume around the robot, given state estimates x_t. We chose to use a voxel grid as our core mapping representation instead of a BEV-based approach in order to enable our system to be more generically used across environments and tasks (e.g. indoor environments, manipulation), which may require 3D information. Furthermore, we are able to generate an accurate BEV representation from a voxel grid, while the opposite is not true. We also find that our voxel mapper's processing times are competitive with an elevation mapping-based baseline and is sufficient for real-time operation. Further details are provided in the appendix. For every timestep in our expert dataset, we are able to represent $O_{0:t}$ with either a voxel map V_t or BEV map B_t containing much of the traversability-relevant information from all prior observations. These maps V_t, B_t will serve as input to the learned components of TravSUITE. For all experiments, our map size is $100m \times 100m \times 20m$ at $0.4m \times 0.4m \times 0.1m$ per cell, centered on the robot. The list of map features can be found in Table I.

Feature	Modality	Rep.	Description
Terrain	Geom.	BEV	Elevation of the terrain surface
Slope	Geom.	BEV	Slope of the terrain surface
Diff	Geom.	BEV	Max elevation over the terrain
min/max z	Geom.	Voxel	The min/max height in each voxel
Seg Logit x16	Seg.	Both	Class score from Talk2Dino [6]
VFM x32	VFM	Both	VFM feature from Radiov3 [46]

TABLE I
List of Mapping Features

1) Generating Hindsight Supervision from Voxel Maps: In order to perform inpainting, we construct an oracular voxel map (M_t^*) and BEV map (B_t^*) for every timestep in our dataset. We do this in a similar fashion to TerrainNet [38] by building a voxel map V_t^* from both past and future sensor measurements. Instead of setting a fixed time window to aggregate (e.g. $V_T^* \sim [V_{t-k} \dots V_{t+k}]$), we instead keep track of the mapping volumes for each timestep and take as V_t^* the aggregate of all voxels from every voxel grid that

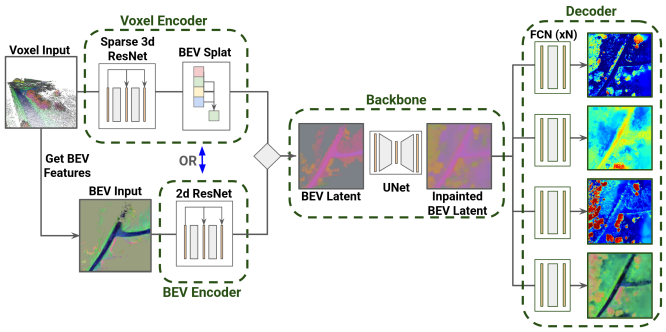

Fig. 3. An overview of our network architecture. The TravSUITE network first starts with a ResNet-style encoder (either 2d or 3d depending on the input modality), followed by BEV-space inpainting and parallel BEV decoder FCN heads. Since the individual prediction heads are relatively lightweight, the backbone and encoder must handle the majority of the feature extraction.

intersects with the volume defined by V_t (detailed algorithm in appendix). This has an advantage in that the quality of aggregation does not depend on robot speed. Additionally, since we directly predict the future map (instead of semantic annotations), this process is fully annotation-free. For both voxel and BEV-based approaches, we choose to only use B^* as a prediction target. Since the supervision is derived from local perception, cells in B^* may still be unreliable at longer sensing distances. To address this, we mask out BEV cells that were observed at a distance no closer than $10m$ from the robot. Note that this distance includes all future poses $x_{0:T}$, meaning B_t^* can contain labels at any distance from the current state x_t and thus requires the network to inpaint.

B. Network Architecture

We instantiate the perception and traversability function from Equation 3 as a multi-headed neural network, similar to ParaDrive [66]. The network takes as input M_t (which may be either V_t or B_t), and uses an encoder and backbone to produce a latent BEV representation Z_t. Multiple different heads are then used to produce navigation-relevant quantities (e.g. costmaps, speedmaps, semantic maps) from Z_t. A high-level architecture diagram is presented in Figure 3.

1) Encoder and Backbone: The role of the encoder and backbone is analogous to the mapping function h_ψ in the previous section, where the goal is to convert the input into a BEV-space feature tensor of a given channel dimension (e.g. $h_\psi(O_{0:t}) \in \mathbb{R}^{F\times W\times H}$). If the input is B_t, this process is simple and can be implemented as a simple ResNet-style [28] network. If the input is V_t, we use a 3D sparse convolution-based analogue using spconv [16] and then splat the 3D features into BEV using a channel-wise max pool.

The backbone of our network is a UNet [49]. This UNet comprises the bulk of the learnable parameters in our network and enables the network to attend to longer-range dependencies in the data. The output of the UNet is a feature tensor $Z \in \mathbb{R}^{F\times W\times H}$ and is used as a shared input across all prediction heads g_ϕ^i.

2) Prediction Heads: In order to convert Z into the same semantic space as B^*, another lightweight decoder FCN

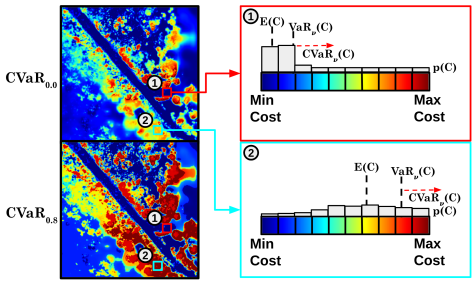

Fig. 4. A diagram of the cost architecture and uncertainty mechanism. (Top-Left) A sample costmap at CVaR=0. (Bottom-Left) The same costmap at CVaR=0.8. (Right) A zoom-in of the cost distribution for two select cells. The red cell has a low-entropy cost distribution and thus stays at a similar value for all CVaR values. The cyan cell has a high-entropy cost distribution and has its cost inflated significantly under higher CVaR.

[35] $g_\phi(Z_t)$ is used. Following TerrainNet's example, we use a separate head for each feature modality (e.g. geometry, semantics). A similar decoder architecture is used to predict cost and speed. However, as we are interested in estimating distributions of cost and speed, we parameterize the outputs of these heads as a categorical distribution over fixed bins. This approach has been demonstrated to be useful when the support of the target variable is known (e.g. traction, speed) [12, 61]. This also enables risk-aware behavior via sampling different risk measures of the resulting distributions (Fig. 4).

C. Leveraging Uncertainty

Since both predicted cost and speed are parameterized as categorical distributions, we are able to evaluate various risk measures such as VaR (quantiles) and CVaR (Fig 4). At test time, we are able to compute a risk-aware cost by evaluating the CVaR of every cell in the costmap. Similarly, we are able to tune the aggressiveness of our speedmap by choosing different quantiles of the resulting speed distributions.

We also implemented a very simple test-time CVaR modulation technique for costs in which a running risk value ν_t was maintained and updated based on the current speed. If the current vehicle speed exceeded a set value, or fell below another set value, the risk tolerance was slightly decremented or incremented, respectively (Equation 8). While simple, this method has the desirable behavior of encouraging safe behavior if the robot is moving quickly, but allowing riskier behavior if little navigation progress is being made.

$$\nu_t = \begin{cases} \nu_{t-1} - \epsilon & v_t < v_{low} \\ \nu_{t-1} + \epsilon & v_t > v_{high} \\ \nu_t & \text{otherwise} \end{cases} \tag{8}$$

D. Training Details

For all experiments, we leverage a dataset of roughly four hours of expert driving at a primary and secondary test site, across all four seasons using an full-scale, autonomous off-road vehicle equipped with the sensor suite from TartanDrive 2.0 [54] (of which we use the LiDAR, IMU and RGB camera). In order to generate state estimates x_t, we leverage an off-the-shelf Lidar-inertial odometry algorithm [72]. Given these

state estimates, we are able to run our VFM voxel mapper to generate inertial-frame VFM voxel maps at $10hz$. The dataset contains roughly 100,000 individual frames with a timestep of $0.1s$, split roughly 80/20 between the two locations. The dataset was divided into a train set and two separate test sets. The **Train** and **Test 1** sets were partitioned (roughly 60/40 by GPS region) from the primary test site, while **Test 2** contains the entirety of the data from the secondary site. Experiments were run for five epochs with a batch size of 3. Since the BEV inpainting targets are multi-modal (e.g. geometric/VFM/semantic features), we follow the procedure used in TerrainNet [38], where each modality is weighted separately. Smooth-L1 [1] ($\beta = 0.2$) is used for continuous features, while cross-entropy is used for segmentation features (Eq. 9).

$$\mathcal{L}_M(B^*, B) = \lambda_{geom} L1(B^*_{geom}, B_{geom}) + L1(\lambda_{VFM}||B^*_{VFM} - B_{VFM}||) + \lambda_{seg} CE(B^*_{seg}, B_{seg}) \quad (9)$$

V. Experiments

In order to evaluate what design decisions matter for downstream path planning, we performed a series of rigorous online and offline evaluations using our dataset and real-world navigation trials.

A. Offline Experiments

In our offline experiments, we answer these questions:

1) **RQ1:** Do auxiliary tasks such as inpainting improve downstream path planning and if so which?
2) **RQ2:** How much does the choice of input representation affect downstream performance?
3) **RQ3:** How do we compare to relevant baselines?

Across our offline experiments, we use a common set of behavior and inpainting metrics. In order to measure the efficacy of the behavioral learning, we report two metrics. Modified Hausdorff distance [18] (MHD) between the expert trajectory and optimal learner trajectory is used to evaluate the cost prediction component. The average probability of the expert speed under the learned speedmap is used to evaluate the speedmap prediction component. To measure inpainting performance, we can directly compute the error between $h_\psi(O_{0:t})$ and B^*_t. We compute error separately for each feature modality as:

1) **Geometry**: L_1 error.
2) **Semantics**: L_1 distance between the semantic distributions (note that neither distribution is one-hot and this essentially measures total variation distance).
3) **VFM**: L_2 error.

B. RQ1: The Effect of Inpainting on Behavior Learning

In order to answer **RQ1**, we trained several variations of our voxel-based network using different permutations of inpainting modalities. In particular, we compare using no inpainting task (**None** in Table II), inpainting only geometric/VFM/segmentation features, and cotraining all inpainting modalities together. Overall we find that across both datasets, inpainting improves both the costmap and speedmap components of our network. Notably, all ablations in this experiment had the same number of behavioral examples, confirming the hypothesis that learning a better map represenation M results in better planning behavior. Perhaps unsurprisingly, using more modalities together seems to result in the best performance, as traversability is a function of both geometry and semantics.

Inpainting	Dataset	MHD ↓	Speed Prob. ↑
All	Test 1	**1.3936**	0.1159
Geometry	Test 1	1.4138	0.1054
VFM	Test 1	1.5400	**0.1162**
Segmentation	Test 1	1.4877	0.1120
None	Test 1	1.7744	0.1026
All	Test 2	**1.3988**	**0.1258**
Geometry	Test 2	1.4410	0.0982
VFM	Test 2	1.4619	0.1158
Segmentation	Test 2	1.4228	0.1004
None	Test 2	1.5400	0.1086

TABLE II
Effect of inpainting on planning tasks

C. RQ2: Choice of Input Representation

In order to answer **RQ2**, we explore several choices of input representations to our method. In particular, we consider three types of inputs:

1) **Voxel**: V_t from Section IV-A.
2) **BEV**: B_t from Section IV-A.
3) **Voxel 1-frame**: V_t, but only containing voxels from the current timestep (to ablate the importance of aggregating observations, as some approaches that inpaint do not aggregate measurements [45, 70], or find performance gains by not doing so [53, 38]).

Similar to the previous section, we report the planning metrics on the held-out dataset in Table III. Additionally, we report the inpainting accuracy in Table IV. Note that the cells used for With respect to path-planning metrics, we observe that the voxel method performs better on the costmap metric while the BEV method is better with respect to the speedmap metric. This indicates that (at least for our environments), both BEV and 3D representations perform reasonably well for traversability. Across both datasets, the single-frame voxel input method performs the worst, suggesting that significant performance gains can be achieved by aggregating sensor measurements.

We find again that aggregation of observations results in better inpainting performance for both observed and unobserved cells. This corroborates the findings of Shaban et al. [53] in that temporal aggregation is helpful in simplifying the map prediction task when odometry is reliable. We observe that with aggregation, both BEV and voxel-based inputs perform similarly, with the BEV-based model performing marginally better. This is likely due to the labels being generated using the same process as the inputs, making the BEV to BEV learning task somewhat simpler.

Input	Dataset	MHD ↓	Speed Prob. ↑
Voxel	Test 1	**1.3936**	0.1159
BEV	Test 1	1.4440	**0.1184**
Voxel no agg.	Test 1	1.5061	0.1109
Voxel	Test 2	**1.3988**	0.1258
BEV	Test 2	1.4373	**0.1311**
Voxel no agg.	Test 2	1.4452	0.1031

TABLE III
EFFECT OF INPUT REPRESENTATION ON PLANNING TASKS

Input	Dataset	Geom ↓	VFM ↓	Seg ↓
Voxel	Test 1	0.7253	1.2324	0.3087
BEV	Test 1	**0.7137**	**1.1450**	**0.3019**
Voxel No Agg.	Test 1	1.5952	1.9533	0.5064
Voxel	Test 2	0.5221	1.2389	0.4279
BEV	Test 2	**0.4704**	**1.1313**	**0.4019**
Voxel No Agg.	Test 2	1.6326	2.0331	0.6033

TABLE IV
EFFECT OF INPUT REPRESENTATION ON INPAINTING TASKS

D. RQ3: Comparison to Other Methods

We also compare our method to several approaches in recent literature. In order to do so, many of the ablations made use of an inpainting-only backbone. This backbone shared the architecture described in the previous section, but was trained only on the inpainting task. **TerrainNet Cfn.**: In order to produce a TerrainNet-style [38] baseline, we used the predicted geometric and semantic maps to implement the hand-designed cost function prescribed by TerrainNet. In order to improve the performance, we fine-tuned the coefficients of the cost function with IRL. As TerrainNet does not produce a speedmap, we do not report a speed metric. **Cascaded Trav.**: We produced a Creste-style [70] baseline by training a FCN with IRL, using the predicted geometric and VFM maps as input. **FCN no Inpaint**: We also trained a Velociraptor-style [61] baseline by training a lightweight FCN on B_t, using only the behavioral objective. **Parallel Trav.** is equivalent to inpainting with all modalities from Table II. Results can be found in Table V.

Overall, we find that both the parallel and cascaded traversability methods outperform the TerrainNet baseline, suggesting that a more complex cost function can lead to improved navigation performance. Interestingly, we find that the FCN baseline (the only method that does not use inpainting) performs the worst across both metrics, suggesting that inpainting with larger architectures is important to learning effective navigation maps. Additionally, we note that the FCN baseline outperforms the larger architecture (no inpainting from Table II), suggesting that it is difficult to train a large-scale architecture with traversability supervision alone. The performance of the parallel and cascaded architecture is relatively comparable with the former producing better speedmaps, and the latter producing better costmaps. Overall, this suggests that learning traversability from demonstrations can outperform traditional baselines, but it is important to add an inpainting task. Some qualitative prediction examples are presented in the hardware experiments section.

Sample qualitative results are presented in Figure 5 in which we compare the voxel network with inpainting (Voxel+All in Table II) to two networks without inpainting (Voxel+None in Table II and FCN no Inpaint from Table V). We observe significant improvements in the stability and accuracy of both the cost and speed maps as a result of inpainting. In practice, this prediction stability is important for real-world navigation.

Approach	Dataset	MHD ↓	Speed Prob. ↑
Parallel Trav.	Test 1	1.3936	**0.1159**
Cascaded Trav.	Test 1	**1.3043**	0.1076
TerrainNet Cfn.	Test 1	1.4553	N/A
FCN no Inpaint	Test 1	1.5246	0.0967
Parallel Trav.	Test 2	1.3988	**0.1258**
Cascaded Trav.	Test 2	**1.3845**	0.1156
TerrainNet Cfn.	Test 2	1.3965	N/A
FCN no Inpaint	Test 2	1.4961	0.1017

TABLE V
COMPARISON TO SIMILAR METHODS

E. Hardware Experiments

We also perform several real-world hardware experiments to answer these questions:

1) Does our approach enable off-road navigation?
2) Does risk-aware planning enable safer navigation?
3) Do our offline findings translate to hardware?

In order to evaluate our algorithm on physical hardware, we set up a navigation trial using our full-scale ATV at our primary test site. The ATV was tasked with navigating as quickly as possible to a series of waypoints designed to force the vehicle to interact with a mixture of on-trail and off-trail scenarios, as well as vegetation and man-made obstacles. The experiment took place in the winter with somewhat significant snow cover. A full traversal of the course was roughly 1.4km.

A more complex version of MPPI was used on hardware. This version of MPPI used a kinematic bicycle model with additional throttle and steering dynamics to model the transient behavior of the throttle and steering actuators on the platform. In addition to the cost function from Equation 2, an additional lateral acceleration limit was added. Given the challenging conditions, the mean predicted speed from the speedmap prediction was used. All mapping and traversability computation was run at roughly $5hz$ using an on-board laptop with an 13th gen Intel i9 CPU and Nvidia 4090 laptop GPU.

Several ablations from the previous section were compared:

1) **Parallel Trav.**: from Section V-D.
2) **TerrainNet Cfn.**: from Section V-D. Note that we also trained a speed prediction head from the inpainted latent features in a similar fashion to the previous sections.
3) **FCN no Inpaint**: from Section V-D.
4) **Parallel Trav. + CVaR**: **Parallel Trav.**, but also using the online CVaR selection method from the Section IV-C

We use the following metrics for evaluation:

1) **Number of Interventions (Int.)**: Number of times a safety operator was required to correct the robot. The safety operator was instructed to intervene if and only if 1) a collision with an obstacle appeared imminent, or 2) the robot stopped making progress towards the next waypoint. If an intervention was called, the safety

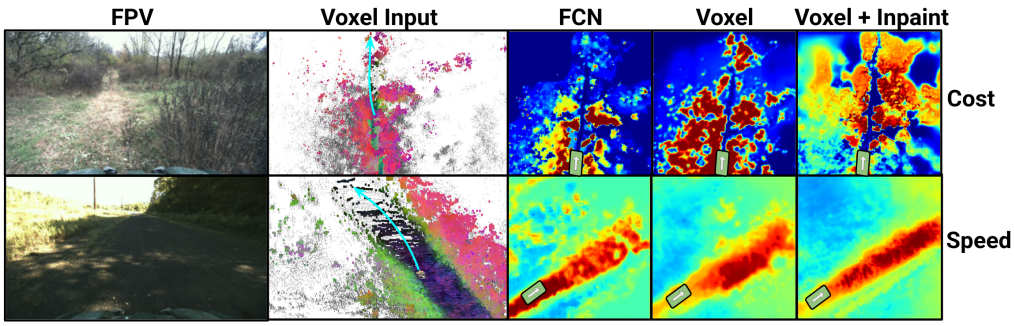

Fig. 5. Two scenarios from the test set. Cost and speed maps are cropped and enlarged to highlight regions of interest. (Top row): The robot must navigate through a path obscured by dense vegetation (cyan arrow). The Voxel+Inpaint network does the best at assigning low cost to the trail and high cost to the surrounding unobserved space, which is likely vegetation. (Bottom row): The robot is traveling at high speed on an open trail. Due to the speed, LiDAR measurements are relatively sparse. The inpainting method is able to fill in the missing trail features and keep a consistent speed map.

operator was instructed to teleoperate the vehicle past the offending obstacle and then resume autonomy.

2) **Number of Collisions (Col.)**: Of the interventions above, the amount that were due to an obstacle collision. We report this number separately as many interventions due to lack of progress can be addressed with better backtracking and planning, while collisions cannot.
3) **Average/Top Auto. Speed**: The average/top speed while autonomous.

F. Analysis of Hardware Results

High-level metrics from each of the autonomy runs are presented in Table VI. A GPS plot of the interventions with FPV views is presented in Figure 6. Overall, we observed that the best-performing method in terms of intervention metrics is **Parallel Trav. + CVaR**. Notably, this was the only method that was able to successfully avoid all of the man-made obstacles, as **Parallel Trav.** and **TerrainNet Cfn.** each had a collision, suggesting that adding risk-awareness results in safer navigation. A comparison of the various methods navigating around some man-made cones can be seen in Figure 8. This scenario required the vehicle to divert from the trail to avoid the cones. Since cones were out-of-distribution from the train set, **Parallel Trav.** was overly optimistic. Similarly, the semantic inpainting struggled to inpaint the cones in the BEV map, leading **TerrainNet Cfn.** to understimate the cost. However, using CVaR to capture the semantic uncertainty enabled the robot to avoid the cones. Additionally, we can see that the **FCN no Inpaint** method was both the slowest and had the largest number of interventions, suggesting that the inpainting had a positive impact on both traversal speed and number of required backtracks. A qualitative example of the value of inpainting is presented in Figure 7 in which the **FCN no Inpaint** method assigned low cost to an area containing vegetation that was out of direct view. Since the goal point was directly beyond this region, the robot mistakenly started planning into this vegetation, resulting in it getting stuck in a cul-de-sac. However, with inpainting, all other models were able to pick up on the surrounding geometric and semantic cues and inferred that the unobserved area was likely high cost, leading the planner to take a longer but safer path on the trail. We also observed that all methods with inpainting had

Method	Col.	Int.	Avg. Speed	Top Speed
Parallel Trav.	2	2	4.79m/s	7.66m/s
Terrainnet Cfn.	1	**1**	5.23m/s	7.72m/s
FCN no Inpaint	1	4	4.33m/s	6.84m/s
Parallel Trav. + CVaR	**0**	**1**	4.49m/s	7.16m/s

TABLE VI
HARDWARE METRICS

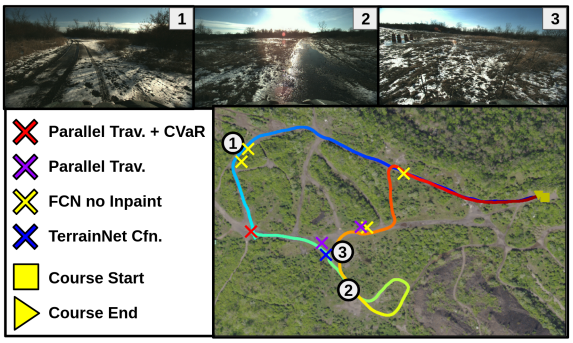

Fig. 6. GPS map of our navigation course. The course totaled about 1.4km in length and contained a number of challenges including snow, vegetation and man-made obstacles. In order to successfully navigate the course, the vehicle must be able to avoid OOD obstacles, drive through tall vegetation, and intelligently select traversal speed based on the terrain surface.

higher average autonomous speed. This is likely because the inpainting objective resulted in more temporally stable speed predictions. Fluctuations in the speedmap, like those seen in Figure 5 force the ATV to reduce its speed not just for the offending cell, but also for prior cells as braking cannot occur instantaneously. Unsurprisingly, adding CVaR (thus reducing risk-tolerance) decreased autonomous speed as the robot had to avoid more objects, but reduced the number of interventions.

VI. CONCLUSION

In this work, we presented TravSUITE, a fully self-supervised traversability and map inpainting learning framework. We demonstrated that inpainting and risk estimation play an important role in learned traversability estimation methods in both offline and hardware results. Furthermore, we have demonstrated a straightforward technique for generating large-scale geometric-semantic inpainting supervision that requires no additional engineering overhead. We believe that this is a significant step forward for self-supervised traversability estimation methods in unstructured terrain.

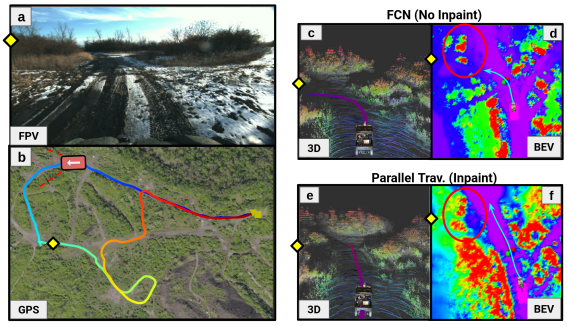

Fig. 7. Qualitative figure from the navigation trial. (a, b) After reaching the current waypoint, the ATV must navigate to another waypoint about $150m$ to its left. (c, d) Without inpainting, the FCN baseline underestimates the cost of the vegetation out of the camera view, resulting in an intervention. (e, f) The inpainting baseline is able to successfully estimate the cost of the vegetation, thus encouraging the local planner to take a longer path on the trail.

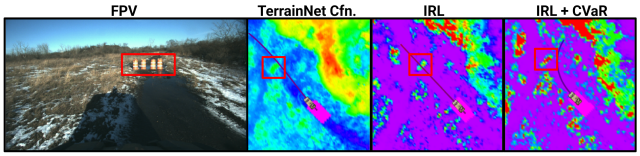

Fig. 8. Qualitative figure from the navigation trial. The robot was tasked with navigating around the set of cones in the trail. (Left) Inpaint + IRL without CVaR plans through the cones. (Center) The TerrainNet cfn does not sufficiently distinguish between the cones and the tall vegetation to its side. (Right) IRL with CVaR assigns enough cost to the cones for the vehicle to plan around into the grass.

While TravSUITE represents step forward for learning-based local traversability, there are some limitations of TravSUITE worth highlighting. First, while this method is environment and robot-generic in principle, additional experiments in additional biomes and platforms are needed to fully substantiate this claim. Second, while we provide a rudimentary uncertainty estimation technique that seems to work well in practice, a more principled treatment of uncertainty (such as [31, 13]) may improve performance in OOD. Lastly, TravSUITE only performs local traversability estimation and cannot reason about information beyond the mapping horizon.

Future work will include exploring additional sources of high-quality self-supervised data and demonstrating our system in more environments such as multi-floor buildings and on other platforms such as legged robots. We are also interested in co-training additional tasks such as full 3d path planning and segmentation that necessitate a volumetric representation. We are also interested in exploring non-metric representations to enable long-range traversability estimation and planning [20, 52].

ACKNOWLEDGMENTS

The authors would like to thank Matthew Sivaprakasam, Micah Nye and Parv Maheshwari for their help with field testing and feedback in the paper-writing stage. This work was supported by DSO National Laboratories (DSO) Contract #DSOCO25020. This work used Bridges-2 at PSC through allocation cis220039p from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program which is supported by NSF grants #2138259, #2138286, #2138307, #2137603, and #213296. This work was supported in part by Field AI, Inc.

References

[1] Pytorch smooth-l1. https://docs.pytorch.org/docs/2.11/generated/torch.nn.SmoothL1Loss.html, 2026.

[2] Kereshmeh Afsari, Srijeet Halder, Mahnaz Ensafi, Stephen DeVito, and John Serdakowski. Fundamentals and prospects of four-legged robot application in construction progress monitoring. *EPiC Series in Built Environment*, 2:274–283, 2021.

[3] Ali Agha, Kyohei Otsu, Benjamin Morrell, David D Fan, Rohan Thakker, Angel Santamaria-Navarro, Sung-Kyun Kim, Amanda Bouman, Xianmei Lei, Jeffrey Edlund, et al. Nebula: Quest for robotic autonomy in challenging environments; team costar at the darpa subterranean challenge. *arXiv preprint arXiv:2103.11470*, 2021.

[4] Omar Alama, Avigyan Bhattacharya, Haoyang He, Seungchan Kim, Yuheng Qiu, Wenshan Wang, Cherie Ho, Nikhil Keetha, and Sebastian Scherer. Rayfronts: Open-set semantic ray frontiers for online scene understanding and exploration. *arXiv preprint arXiv:2504.06994*, 2025.

[5] James Andrew Bagnell, David Bradley, David Silver, Boris Sofman, and Anthony Stentz. Learning for autonomous navigation. *IEEE Robotics & Automation Magazine*, 17(2):74–84, 2010.

[6] Luca Barsellotti, Lorenzo Bianchi, Nicola Messina, Fabio Carrara, Marcella Cornia, Lorenzo Baraldi, Fabrizio Falchi, and Rita Cucchiara. Talking to dino: Bridging self-supervised vision backbones with language for open-vocabulary segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22025–22035, 2025.

[7] C Dario Bellicoso, Marko Bjelonic, Lorenz Wellhausen, Kai Holtmann, Fabian Günther, Marco Tranzatto, Peter Fankhauser, and Marco Hutter. Advances in real-world applications for legged robots. *Journal of Field Robotics*, 35(8):1311–1326, 2018.

[8] Paulo Borges, Thierry Peynot, Sisi Liang, Bilal Arain, Matthew Wildie, Melih Minareci, Serge Lichman, Garima Samvedi, Inkyu Sa, Nicolas Hudson, et al. A survey on terrain traversability analysis for autonomous ground vehicles: Methods, sensors, and challenges. *Field Robotics*, 2(1):1567–1627, 2022.

[9] Amanda Bouman, Muhammad Fadhil Ginting, Nikhilesh Alatur, Matteo Palieri, David D Fan, Thomas Touma, Torkom Pailevanian, Sung-Kyun Kim, Kyohei Otsu, Joel Burdick, et al. Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2518–2525. IEEE, 2020.

[10] Matěj Boxan, Alexander Krawciw, Effie Daum, Xinyuan Qiao, Sven Lilge, Timothy D Barfoot, and François Pomerleau. Fomo: A proposal for a multi-season dataset for robot navigation in for\^ et montmorency. *arXiv preprint arXiv:2404.13166*, 2024.

[11] Xiaoyi Cai, Michael Everett, Jonathan Fink, and Jonathan P How. Risk-aware off-road navigation via a learned speed distribution map. *arXiv preprint arXiv:2203.13429*, 2022.

[12] Xiaoyi Cai, Siddharth Ancha, Lakshay Sharma, Philip R Osteen, Bernadette Bucher, Stephen Phillips, Jiuguang Wang, Michael Everett, Nicholas Roy, and Jonathan P How. Evora: Deep evidential traversability learning for risk-aware off-road autonomy. *arXiv preprint arXiv:2311.06234*, 2023.

[13] Xiaoyi Cai, James Queeney, Tong Xu, Aniket Datar, Chenhui Pan, Max Miller, Ashton Flather, Philip R Osteen, Nicholas Roy, Xuesu Xiao, et al. Pietra: Physics-informed evidential learning for traversing out-of-distribution terrain. *arXiv preprint arXiv:2409.03005*, 2024.

[14] Mateo Guaman Castro, Samuel Triest, Wenshan Wang, Jason M Gregory, Felix Sanchez, John G Rogers III, and Sebastian Scherer. How does it feel? self-supervised costmap learning for off-road vehicle traversability. *arXiv preprint arXiv:2209.10788*, 2022.

[15] Annett Chilian and Heiko Hirschmüller. Stereo camera based navigation of mobile robots on rough terrain. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4571–4576. IEEE, 2009.

[16] Spconv Contributors. Spconv: Spatially sparse convolution library. https://github.com/traveller59/spconv, 2022.

[17] Anushri Dixit, David D Fan, Kyohei Otsu, Sharmita Dey, Ali-Akbar Agha-Mohammadi, and Joel Burdick. Step: Stochastic traversability evaluation and planning for risk-aware navigation; results from the darpa subterranean challenge. *Field Robotics*, 4:182–210, 2024.

[18] M-P Dubuisson and Anil K Jain. A modified hausdorff distance for object matching. In *Proceedings of 12th international conference on pattern recognition*, volume 1, pages 566–568. IEEE, 1994.

[19] Gian Erni, Jonas Frey, Takahiro Miki, Matias Mattamala, and Marco Hutter. Mem: Multi-modal elevation mapping for robotics and learning. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11011–11018. IEEE, 2023.

[20] Ethan Fahnestock, Erick Fuentes, Samuel Prentice, Vasileios Vasilopoulos, Philip R Osteen, Thomas Howard, and Nicholas Roy. Far-field image-based traversability mapping for a priori unknown natural environments. *IEEE Robotics and Automation Letters*, 2025.

[21] David D Fan, Kyohei Otsu, Yuki Kubo, Anushri Dixit, Joel Burdick, and Ali-Akbar Agha-Mohammadi. Step: Stochastic traversability evaluation and planning for risk-aware off-road navigation. *arXiv preprint arXiv:2103.02828*, 2021.

[22] Péter Fankhauser and Marco Hutter. A universal grid map library: Implementation and use case for rough terrain navigation. In *Robot Operating System (ROS)*, pages 99–120. Springer, 2016.

[23] Péter Fankhauser, Marko Bjelonic, C Dario Bellicoso, Takahiro Miki, and Marco Hutter. Robust rough-terrain

locomotion with a quadrupedal robot. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5761–5768. IEEE, 2018.

[24] Jonas Frey, David Hoeller, Shehryar Khattak, and Marco Hutter. Locomotion policy guided traversability learning using volumetric representations of complex environments. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5722–5729. IEEE, 2022.

[25] Jonas Frey, Matias Mattamala, Nived Chebrolu, Cesar Cadena, Maurice Fallon, and Marco Hutter. Fast traversability estimation for wild visual navigation. *arXiv preprint arXiv:2305.08510*, 2023.

[26] Jonas Frey, Shehryar Khattak, Manthan Patel, Deegan Atha, Julian Nubert, Curtis Padgett, Marco Hutter, and Patrick Spieler. Roadrunner–learning traversability estimation for autonomous off-road driving. *arXiv preprint arXiv:2402.19341*, 2024.

[27] Tianrui Guan, Divya Kothandaraman, Rohan Chandra, Adarsh Jagan Sathyamoorthy, Kasun Weerakoon, and Dinesh Manocha. Ga-nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments. *IEEE Robotics and Automation Letters*, 7(3):8138–8145, 2022.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[29] Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripalli. Rellis-3d dataset: Data, benchmarks and analysis, 2020.

[30] Sanghun Jung, JoonHo Lee, Xiangyun Meng, Byron Boots, and Alexander Lambert. V-strong: Visual self-supervised traversability learning for off-road navigation. *arXiv preprint arXiv:2312.16016*, 2023.

[31] Sanghun Jung, Daehoon Gwak, Byron Boots, and James Hays. Uncertainty-aware accurate elevation modeling for off-road navigation via neural processes. *arXiv preprint arXiv:2508.03890*, 2025.

[32] Haresh Karnan, Elvin Yang, Daniel Farkash, Garrett Warnell, Joydeep Biswas, and Peter Stone. Sterling: Self-supervised terrain representation learning from unconstrained robot experience. In *7th Annual Conference on Robot Learning*, 2023.

[33] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[34] Philipp Krüsi, Paul Furgale, Michael Bosse, and Roland Siegwart. Driving on point clouds: Motion planning, trajectory optimization, and terrain assessment in generic nonplanar environments. *Journal of Field Robotics*, 34 (5):940–984, 2017.

[35] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[36] Daniel Maturana. *Semantic Mapping for Autonomous Navigation and Exploration*. PhD thesis, Carnegie Mellon University, 2022.

[37] Daniel Maturana, Po-Wei Chou, Masashi Uenoyama, and Sebastian Scherer. Real-time semantic mapping for autonomous off-road navigation. In *Field and Service Robotics*, pages 335–350. Springer, 2018.

[38] Xiangyun Meng, Nathan Hatch, Alexander Lambert, Anqi Li, Nolan Wagener, Matthew Schmittle, JoonHo Lee, Wentao Yuan, Zoey Chen, Samuel Deng, et al. Terrainnet: Visual modeling of complex terrain for high-speed, off-road navigation. *arXiv preprint arXiv:2303.15771*, 2023.

[39] Isaac Miller, Sergei Lupashin, Noah Zych, Pete Moran, Brian Schimpf, Aaron Nathan, and Ephrahim Garcia. Cornell university's 2005 darpa grand challenge entry. *Journal of Field Robotics*, 23(8):625–652, 2006.

[40] Peter Mortimer, Raphael Hagmanns, Miguel Granero, Thorsten Luettel, Janko Petereit, and Hans-Joachim Wuensche. The goose dataset for perception in unstructured environments. URL https://arxiv.org/abs/2310.16788.

[41] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[42] Timothy Overbye and Srikanth Saripalli. G-vom: A gpu accelerated voxel off-road mapping system. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 1480–1486. IEEE, 2022.

[43] Manthan Patel, Jonas Frey, Deegan Atha, Patrick Spieler, Marco Hutter, and Shehryar Khattak. Roadrunner m&m-learning multi-range multi-resolution traversability maps for autonomous off-road navigation. *IEEE Robotics and Automation Letters*, 2024.

[44] Patrick Pfaff, Rudolph Triebel, and Wolfram Burgard. An efficient extension to elevation maps for outdoor terrain mapping and loop closing. *The International Journal of Robotics Research*, 26(2):217–230, 2007.

[45] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020.

[46] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative model–reduce all domains into one. *arXiv preprint arXiv:2312.06709*, 2023.

[47] Nathan D Ratliff, David Silver, and J Andrew Bagnell. Learning to search: Functional gradient techniques for

imitation learning. *Autonomous Robots*, 27(1):25–53, 2009.

[48] R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2: 21–42, 2000.

[49] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[50] Sebastian Scherer, Vasu Agrawal, Graeme Best, Chao Cao, Katarina Cujic, Ryan Darnley, Robert DeBortoli, Eric Dexheimer, Bill Drozd, Rohit Garg, Ian Higgins, John Keller, David Kohanbash, Lucas Nogueira, Roshan Pradhan, Michael Tatum, Vaibhav K. Viswanathan, Steven Willits, Shibo Zhao, Hongbiao Zhu, Dan Abad, Tim Angert, Greg Armstrong, Ralph Boirum, Adwait Dongare, Matthew Dworman, Shengjie Hu, Joshua Jaekel, Ran Ji, Alice Lai, Yu Hsuan Lee, Anh Luong, Joshua Mangelson, Jay Maier, James Picard, Kevin Pluckter, Andrew Saba, Manish Saroya, Emily Scheide, Nathaniel Shoemaker-Trejo, Joshua Spisak, Jim Teza, Fan Yang, Andrew Wilson, Henry Zhang, Howie Choset, Michael Kaess, Anthony Rowe, Sanjiv Singh, Ji Zhang, Geoffrey A. Hollinger, and Matthew Travers. Resilient and modular subterranean exploration with a team of roving and flying robots. *Field Robotics Journal*, pages 678–734, May 2022.

[51] Robin Schmid, Deegan Atha, Frederik Schöller, Sharmita Dey, Seyed Fakoorian, Kyohei Otsu, Barry Ridge, Marko Bjelonic, Lorenz Wellhausen, Marco Hutter, et al. Self-supervised traversability prediction by learning to reconstruct safe terrain. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12419–12425. IEEE, 2022.

[52] Matt Schmittle, Rohan Baijal, Nathan Hatch, Rosario Scalise, Mateo Guaman Castro, Sidharth Talia, Khimya Khetarpal, Byron Boots, and Siddhartha Srinivasa. Long range navigator (lrn): Extending robot planning horizons beyond metric maps. *arXiv preprint arXiv:2504.13149*, 2025.

[53] Amirreza Shaban, Xiangyun Meng, JoonHo Lee, Byron Boots, and Dieter Fox. Semantic terrain classification for off-road autonomous driving. In *Conference on Robot Learning*, pages 619–629. PMLR, 2022.

[54] Matthew Sivaprakasam, Parv Maheshwari, Mateo Guaman Castro, Samuel Triest, Micah Nye, Steve Willits, Andrew Saba, Wenshan Wang, and Sebastian Scherer. Tartandrive 2.0: More modalities and better infrastructure to further self-supervised learning research in off-road driving tasks. *arXiv preprint arXiv:2402.01913*, 2024.

[55] Matthew Sivaprakasam, Samuel Triest, Cherie Ho, Shubhra Aich, Jeric Lew, Isaiah Adu, Wenshan Wang, and Sebastian Scherer. Salon: Self-supervised adaptive learning for off-road navigation. *arXiv preprint arXiv:2412.07826*, 2024.

[56] Maximilian Stölzle, Takahiro Miki, Levin Gerdes, Martin Azkarate, and Marco Hutter. Reconstructing occluded elevation information in terrain maps with self-supervised learning. *IEEE Robotics and Automation Letters*, 7(2): 1697–1704, 2022.

[57] Stanford Racing Team. Stanford racing team's entry in the 2005 darpa grand challenge. *Published on DARPA Grand Challenge*, 2005.

[58] Marco Tranzatto, Takahiro Miki, Mihir Dharmadhikari, Lukas Bernreiter, Mihir Kulkarni, Frank Mascarich, Olov Andersson, Shehryar Khattak, Marco Hutter, Roland Siegwart, et al. Cerberus in the darpa subterranean challenge. *Science Robotics*, 7(66):eabp9742, 2022.

[59] Samuel Triest, Mateo Guaman Castro, Parv Maheshwari, Matthew Sivaprakasam, Wenshan Wang, and Sebastian Scherer. Learning risk-aware costmaps via inverse reinforcement learning for off-road navigation. *arXiv preprint arXiv:2302.00134*, 2023.

[60] Samuel Triest, David D Fan, Sebastian Scherer, and Ali-Akbar Agha-Mohammadi. Unrealnet: Learning uncertainty-aware navigation features from high-fidelity scans of real environments. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12627–12634. IEEE, 2024.

[61] Samuel Triest, Matthew Sivaprakasam, Shubhra Aich, David Fan, Wenshan Wang, and Sebastian Scherer. Velociraptor: Leveraging visual foundation models for label-free, risk-aware off-road navigation. In *8th Annual Conference on Robot Learning*, 2024.

[62] Cris Urmson, Joshua Anhalt, Michael Clark, Tugrul Galatali, Juan Pablo Gonzalez, Jay Gowdy, Alexander Gutierrez, Sam Harbaugh, Matthew Johnson-Roberson, Hiroki Kato, et al. High speed navigation of unrehearsed terrain: Red team technology for grand challenge 2004. *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-04-37*, 1, 2004.

[63] Kasun Weerakoon, Mohamed Elnoor, Gershom Seneviratne, Vignesh Rajagopal, Senthil Hariharan Arul, Jing Liang, Mohamed Khalid M Jaffar, and Dinesh Manocha. Behav: Behavioral rule guided autonomy using vlms for robot navigation in outdoor scenes. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7044–7051. IEEE, 2025.

[64] Lorenz Wellhausen, Alexey Dosovitskiy, René Ranftl, Krzysztof Walas, Cesar Cadena, and Marco Hutter. Where should i walk? predicting terrain properties from images via self-supervised learning. *IEEE Robotics and Automation Letters*, 4(2):1509–1516, 2019.

[65] Carl Wellington, Aaron C Courville, and Anthony Stentz. Interacting markov random fields for simultaneous terrain modeling and obstacle detection. In *Robotics: Science and Systems*, volume 6, pages 1–8, 2005.

[66] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 15449–15458, 2024.

[67] Maggie Wigness, John G Rogers, and Luis E Navarro-Serment. Robot navigation from human demonstration: Learning control behaviors. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1150–1157. IEEE, 2018.

[68] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information theoretic mpc for model-based reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation*, pages 1714–1721. IEEE, 2017.

[69] Stuart H Young. Robot autonomy in complex environments. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III*, volume 11746, page 1174602. SPIE, 2021.

[70] Arthur Zhang, Harshit Sikchi, Amy Zhang, and Joydeep Biswas. Creste: Scalable mapless navigation with internet scale priors and counterfactual guidance. *arXiv preprint arXiv:2503.03921*, 2025.

[71] Yanfu Zhang, Wenshan Wang, Rogerio Bonatti, Daniel Maturana, and Sebastian Scherer. Integrating kinematics and environment context into deep inverse reinforcement learning for predicting off-road vehicle trajectories. *arXiv preprint arXiv:1810.07225*, 2018.

[72] Shibo Zhao, Hengrui Zhang, Peng Wang, Lucas Nogueira, and Sebastian Scherer. Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8729–8736. IEEE, 2021.

[73] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.