Far-Field Image-Based Traversability Mapping for *A Priori* Unknown Natural Environments

Ethan Fahnestock[1], Erick Fuentes[1], Samuel Prentice[1], Vasileios Vasilopoulos[2], Philip R Osteen[3], Thomas Howard[4], Nicholas Roy[1]

***Abstract*—While navigating unknown environments, robots rely primarily on proximate features for guidance in decision making, such as depth information from lidar or stereo to build a costmap, or local semantic information from images. The limited range over which these features can be used may result in poor robot behavior when assumptions about the cost of the map beyond the range of proximate features misguide the robot. Integrating "far-field" image features that originate beyond these proximate features into the mapping pipeline has the promise of enabling more intelligent and aware navigation through unknown terrain. To navigate with far-field features, key challenges must be overcome. As far-field features are typically too distant to localize precisely, they are difficult to place in a map. Additionally, the large distance between the robot and these features makes connecting these features to their navigation implications more challenging. We propose *FITAM*, an approach that learns to use far-field features to predict costs to guide navigation through unknown environments from previous experience in a self-supervised manner. Unlike previous work, our approach does not rely on flat ground plane assumptions or range sensors to localize observations. We demonstrate the benefits of our approach through simulated trials and real-world deployment on a Clearpath Robotics Warthog navigating through a forest environment. Code is available at github.com/efahnestock/fitam.**

I. INTRODUCTION

When faced with the challenge of navigating an unmapped environment, robots traditionally rely on the guidance of a global planner to reach a goal location. Without a prior map, global planners must make decisions using the robot's own accumulated sensor data. Ranging sensors like lidar or depth cameras provide highly localized information to the robot about its surroundings and returns from these sensors quickly become sparse far from the robot. The sparsity of long-range sensor data limits the robot's ability to estimate the cost (time, energy, risk, etc.) beyond a distance we refer to as the "costmap horizon". Beyond the costmap horizon, the global planner must weigh the cost of traveling through unknown space against the cost of traveling through previously observed space. Global motion planners traditionally assign a fixed cost to unknown space [1]. When this constant cost differs significantly from the true cost of moving through the unknown region, global navigation can mislead the robot into taking sub-optimal paths.

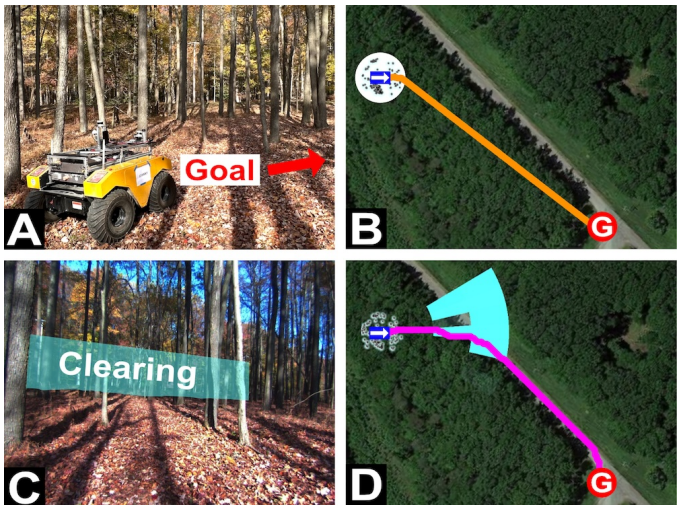

Fig. 1. **A:** A robot is tasked with navigating to a goal far off to the right. **B:** Traditional mapping architectures direct the robot (blue rectangle) down the orange path to its goal (red circle), as its range sensors cannot build a map with a costmap horizon (white circle) that contains a more efficient path. **C:** However, the robot's camera image reveals a clearing, offering the possibility of terrain on which the robot can reach its objective faster. **D:** Our approach *FITAM* learns to use distant visual features for mapping. With *FITAM*, the robot incorporates the low cost predictions (cyan) of the road into its costmap and plans the purple path that leverages the clearing to reach its goal in a faster manner.

If a satellite image or other useful navigation prior is available, it can be used to inform the global planner beyond the costmap horizon [2]. However, relying on these kinds of map priors limits the deployment of robots in cases of rapid environment change (e.g., natural disasters), dynamic environments, GPS denied environments (where localizing within the map prior is difficult), and when the ground is occluded in the overhead image (e.g., tree cover).

Consider the example shown in Figure 1. A robot is tasked with traveling far off to its right. It sits in a forest, but straight ahead—beyond its costmap horizon (white circle in B)—is an easy-to-traverse road. Lacking this road in its map, the robot will plan to take the orange path where navigating through the dense forest would ultimately be slower and more expensive than taking the road. Though the road lies beyond the costmap horizon, the camera image in Figure 1C reveals a clearing through the trees to the left that signifies the presence of the

Research was sponsored by the Army Research Office and was accomplished under Cooperative Agreement Number W911NF-21-2-0150. 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 Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. The lead author was supported by the Department of Defense through the National Defense Science & Engineering Graduate Fellowship Program.

[1]Computer Science and Artificial Intelligence Laboratory (CSAIL), MIT, Cambridge, MA 02139. [2]Ghost Robotics Corporation [3]DEVCOM Army Research Laboratory [4]University of Rochester. E-mail: ekf@mit.edu.

road and a lower cost path. Unlike ranging sensors, images capture rich and dense information about distant objects. In this work, we leverage visual information in images (e.g., the view of the clearing) to provide additional guidance to the robot beyond its costmap horizon. We refer to image features that are useful for navigation that may originate outside of the robot's costmap horizon as "far-field" features.

A key challenge in leveraging far-field features is that images do not provide direct range information, which makes using these kinds of features difficult for planning. Previous work addresses this challenge by either projecting the features onto data from ranging sensors, which inherits the limitations of the ranging sensors [3]–[5], or by assuming a flat ground plane [6]–[8], which works poorly with occlusions and suffers from large projection errors at far distances. Developments in deep learning have produced richer features enabling monocular depth estimation, but accuracy decays at larger ranges (e.g. $\pm 13\mathrm{m}$ at $80\mathrm{m}$) [9], [10]. Our insight is that these richer features can be used to coarsely estimate costs at distant ranges even if the features themselves cannot be localized directly in the map. This insight allows us to incorporate far-field features into our costmaps at distances up to $100\mathrm{m}$ in real-world trials without relying on ranging sensors or a flat ground plane assumption.

To take a step towards leveraging far-field features independently of ranging sensors, we propose *FITAM*: Far-field Image-based TraversAbility Mapping. Our approach efficiently learns from past navigation data in a self-supervised manner to link distant image features with navigation costs at range. The main contributions of this work are as follows:

- A method for learning correlations between far-field features and cost at range in a self-supervised manner.
- Demonstrated improvement over prior work leveraging far-field information for large simulated navigation tasks.
- Deployment of *FITAM* on a physical platform for experiments in a road-forest environment.

II. Related Works

Estimating the traversability of local terrain from images has been the focus of many papers [11], with works exploring learning traversability from demonstration [12], inverse reinforcement learning [13], supervised learning [14], and self-supervised learning [7], [15], [16] with labels generated automatically from proprioceptive sensors like an IMU. These traversability estimates are either learned directly or projected onto a local map to be used for motion planning.

When maximizing the range of these traversability estimates, a number of works cleverly project image-informed traversability estimates with stereo [4], [5] or lidar [17]. However, these approaches inherit the limitations of the ranging sensors, which limits the range of the traversability estimates, particularly in the case of stereo. Works that do not rely on ranging sensors project image-space traversability classification directly onto a ground plane [6]–[8]. This projection enables use of more distant far-field features for planning, but does not generalize well to non-flat or occlusion heavy outdoor environments, as small changes in orientation or non-flat terrains result in large projection errors. In our work we build upon previous image-only long range traversability estimation works by removing the flat ground plane assumption, predicting ranges of cost values instead of just traversable/untraversable, and by predicting costs at distances greater than $50\mathrm{m}$.

In [18], super-pixels extend locally traversable patches into the far-field, but this relies on visual similarity and was not deployed on a robot. In [19], semantic images are used to improve navigation through offline reinforcement learning (RL). In our work, we predict costs into maps instead of predicting costs of short 4-7 second trajectories. In [20], the authors reduce segmented images of urban areas into road networks to support global planning past the local sensor horizon for UAVs. This leverages far-field features to bias planning, but does so in a way overfit to the road networks present in structured, urban environments. In our work, we target unstructured natural terrains. In [21] the authors use panoramas to help predict the likelihood of success and cost of high-level actions which is used to guide long-horizon planning. It is not obvious how navigation in outdoor natural environments would be distilled into a limited set of high-level actions.

III. Problem Formulation

The problem of navigation in unknown environments can be posed as a constrained optimization problem. A robot aims to travel from a start state $\mathbf{x}_s \in \mathcal{S}$ to a goal state $\mathbf{x}_g \in \mathcal{S}$ along a trajectory $\tau(t) : [t_0, t_f] \to \mathcal{S}$, where $t_0, t_f \in \mathbb{R}_+$ and $\mathcal{S}$ is the state space. The trajectory should minimize a cost $c_{\text{traj}}(\tau) : \mathbb{T} \to \mathbb{R}_+$ where $\mathbb{T}$ is the space of all trajectories. This cost function is often decomposed as a sum over state costs $c_{\text{state}}(\mathbf{x}) : \mathcal{S} \to \mathbb{R}_+$ along the trajectory $c_{\text{traj}}(\tau) = \int_{t_0}^{t_f} c_{\text{state}}(\tau(t))\delta t$. The constrained optimization problem is stated below.

$$\begin{aligned} \tau^* &= \arg\min_{\tau \in \mathbb{T}} c_{\text{traj}}(\tau) \\ \tau(t_0) &= \mathbf{x}_s \\ \tau(t_f) &= \mathbf{x}_g \end{aligned} \tag{1}$$

In *a priori* unknown environments, solving this optimization problem is impossible since the state cost function $c_{\text{state}}(\mathbf{x})$ is unknown for most states, and is only revealed as the robot travels through the environment. Limited observability splits the state space $\mathcal{S}$ into two sets: unobserved space $\mathbf{\Omega} \subset \mathcal{S}$, for which the cost assignment is uninformed and fixed at a constant value, and observed space $\mathbf{\Lambda} = \mathcal{S} - \mathbf{\Omega}$, where an estimated mapping $c_{\text{state}}(\mathbf{x})$ has been obtained. As the robot moves to a new state $\mathbf{x}_t$, the set of states $\lambda \subset \mathcal{S}$ for which the cost $c_{\text{state}}(\mathbf{x})$ can be estimated is defined by the state observation function $\mathbf{o}(\mathbf{x}_t) : \mathcal{S} \to 2^{\mathcal{S}}$. States in λ that were previously unobserved transition from $\mathbf{\Omega}$ to $\mathbf{\Lambda}$, while the cost estimate $c_{\text{state}}(\mathbf{x})$ is updated for all states in λ.

In this work, we propose a method to expand the coverage of the observation function $\tilde{\mathbf{o}}(\mathbf{x}_t) \supset \mathbf{o}(\mathbf{x}_t)$ through the use of

far-field features. By providing better observation coverage, we improve the solutions found to the constrained optimization problem posed in Equation 1, and thus improve navigation performance through unknown environments.

IV. METHODS

We propose an approach that learns to predict distant costs at range from local RGB camera data. This information is used to supplement costmaps produced by traditional methods by filling in unobserved costs beyond the costmap horizon. Our approach enables in-the-field training in a self-supervised manner from autonomous or teleoperated data collects, reducing the overhead of operating in new environments.

A. Learning to Predict Cost at Range

Our goal is to predict cost at range. Our planning objective is minimizing time-to-goal, thus the cost of an area is the time to traverse it. While monocular images do not directly encode the distance to objects, they do capture the objects' bearing. To leverage this information, we use a polar coordinate discretization of the state space centered on the robot.

More specifically, we define a discretization of the state space, or "bin", to span an angle $\Delta\theta_b \in [0, 2\pi]$ while lying between two radii r_j and r_{j+1} in a polar coordinate system (see Figure 2). These bins cover the space around the robot radially with boundaries defined by $\mathbf{r}_b = (r_0, \ldots, r_{B+1}), r_j \in \mathbb{R}^+, r_j < r_{j+1}$. The number of bins along the radial dimension is $B = \dim(\mathbf{r}_b) - 1$. A set of radial bins refers to the B bins at a given angle as seen in Figure 2. We also discretize the cost space into a fixed number G of "cost classes". In our application we use time per unit distance ($\mathrm{s\,m^{-1}}$) as cost. Thus, the cost space classes span the speeds of the robot.

To match the polar discretization, RGB images $\mathbf{I}_t \in [0,1]^{H\times W\times 3}$ with height H and width W from the robot are split into K vertical slices $\mathbf{I}_t^{(i)} \in [0,1]^{H\times W'\times 3}, i \in [K]$ where each slice of width W' pixels has a horizontal field-of-view (FOV) matching $\Delta\theta_b$, and $[K]$ refers to $0, \ldots, K-1$. Our objective is to train an ensemble of M models for each bin $j \in [B]$ along the radial direction. Each ensemble of models takes features $\mathbf{f}_t^{(i)} \in \mathbb{R}^D$ extracted from an image slice $\mathbf{I}_t^{(i)}$ and produces a categorical distribution $\mathbf{P}_t^{(i,j)}$ over cost classes[1] as shown in Figure 2.

B. Automatic Far-Field Cost Labeling

Like previous work [22], [23], we employ near-to-far learning to connect local observations with distant images of the terrain by labeling robot experience in a self-supervised manner. Our approach consumes robot navigation experience to learn to leverage far-field features for navigation. We assume that the data consists of RGB images $\mathbf{I}_t$ and lidar scans $\mathbf{L}_t \in \mathbb{R}^{3\times P}$ where P is the number of points. Note that the lidar data are only used at training time and are not used at test time to produce a cost estimate. We use lidar SLAM [24] to produce odometry (position and orientation) relative to the starting state $\mathbf{x}_s \in \mathrm{SE}(3)$.

[1]For this task we found classification generally outperformed regression.

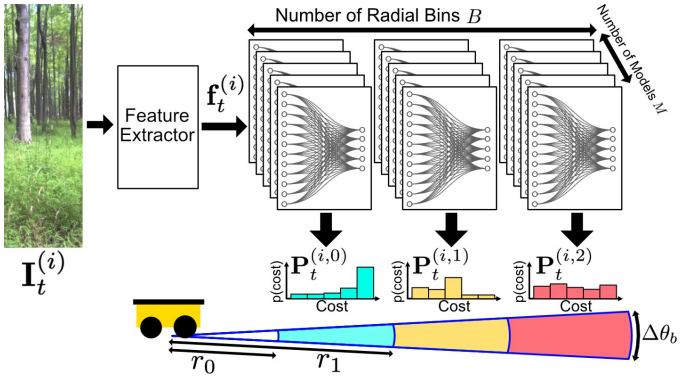

Fig. 2. The structure of the learning problem.

The first step is building a 2D costmap from semantics. We segment each image with FC-HarDNet [25], and map the semantic class of each segment to a cost (reasonable traversal time). We project these cost segments from the images onto lidar scans to build an accumulated map of local cost estimates $\mathbf{m}_l : p \to \mathbb{R}_+$ which maps an observed position $p \in \mathbb{R}^2$ to the cost in $\mathrm{s\,m^{-1}}$ the robot is expected to accrue at that position. A visualization of the semantic map produced by this process (before converting semantics to cost) is overlaid on the environment in both Figure 3 and Figure 6.

Each image $\mathbf{I}_t$ is split into slices $\mathbf{I}_t^{(i)}$ that become the input to the model. Each image slice $\mathbf{I}_t^{(i)}$ has a horizontal FOV matching the angular bin width $\Delta\theta_b$. The number of image slices is set to $K = \lceil \frac{\mathrm{fov}(\mathbf{I}_t)}{\Delta\theta_b} \rceil$ where $\mathrm{fov}(\mathbf{I}_t)$ is the FOV of $\mathbf{I}_t$. To create the supervised labels for each input image slice $\mathbf{I}_t^{(i)}$, the respective set of radial bins is overlaid on the local costmap $\mathbf{m}_l$ using the current robot's state $\mathbf{x}_t$ and the FOV of the image slice. This is visualized in Figure 3.

To emphasize useful terrain, the lowest cost class that is present in at least ρ_{bin} percent of the cells within the jth radial bin ($j \in [B]$) is assigned as the label, $l_t^{(i,j)} \in [G]$ for that bin. Additionally, a weight $w_t^{(i,j)} \in [0,1]$ equal to the percentage of the cells that have been observed within that bin is calculated for use during training to filter ambiguous data. If no class covers more than $\rho_{\text{bin}}\%$ of the radial bin, no label is assigned and the image slice is discarded for that bin's dataset. After the weights and labels are produced for every image slice, a dataset $\mathcal{D}_{\text{ff,j}} = \{(\mathbf{I}_t^{(i)}, l_t^{(i,j)}, w_t^{(i,j)}) | t \in [T], i \in [K]\}$ is produced for each bin $j \in [B]$. To reduce class imbalance a final dataset $\mathcal{D}_{\text{ff,j}}^{\text{bal}} \subseteq \mathcal{D}_{\text{ff,j}}$ is calculated by randomly sampling $N \leq \rho_{\text{class}} \min_{c\in[G]} \mathrm{count}(c)$ tuples for each class c, where $\mathrm{count}(c)$ is the count of class c in $\mathcal{D}_{\text{ff,j}}$ and ρ_{class} fixes the maximum ratio of class counts. This final dataset $\mathcal{D}_{\text{ff,j}}^{\text{bal}}$ is used to train the far-field model for bin j.

C. Model Architecture and Training

The model architecture is shown in Figure 2. Features $\mathbf{f}_t^{(i)} \in \mathbb{R}^D$ of dimension D are extracted from the input image slice $\mathbf{I}_t^{(i)}$ with either a frozen pre-trained network (e.g., DinoV2 [26]) or a jointly trained network. Motivated by increasing uncertainty on out-of-distribution data [27], an ensemble of M single, fully-connected layers are used to map this feature vector $\mathbf{f}_t^{(i)}$ to M categorical distributions over cost classes

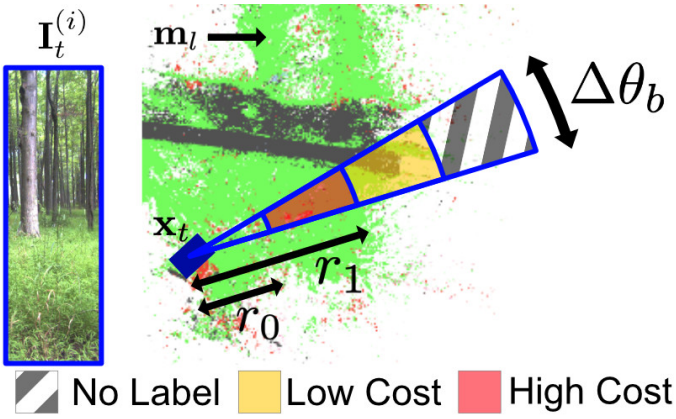

Fig. 3. Label and weight generation for one image slice $\mathbf{I}_t^{(i)}$. The B radial bins are overlaid on the local costmap $\mathbf{m}_l$.

for each bin. These distributions are then averaged, increasing uncertainty in cases of model disagreement, producing a final categorical distribution $\mathbf{P}_t^{(i,j)}$ over cost classes for each bin.

To promote diversity in the ensemble member's outputs, each network $m \in [0, M]$ of the ensemble is trained on its own bootstrap sampled dataset $\mathcal{D}^{\text{m}}_{\text{boot,j}} \subset \mathcal{D}^{\text{bal}}_{\text{ff,j}}$. The fraction of $\mathcal{D}^{\text{bal}}_{\text{ff,j}}$ sampled is fixed at ρ_{boot}. For each bootstrapped dataset a class weight is calculated $\gamma_j^c = \sum_{(\mathbf{I}_t^{(i)}, l_t^{(i,j)}, w_t^{(i,j)}) \in \mathcal{D}^{\text{m}}_{\text{boot,j}}} \mathbb{I}(l_t^{(i,j)} = c)/|\mathcal{D}^{\text{m}}_{\text{boot,j}}|$, where $\mathbb{I}$ is the indicator function. A Cross Entropy (CE) loss weighted by $w_t^{(i,j)} \gamma_j^{l_t^{(i,j)}}$ is used to emphasize accurate prediction of radial bins that have good coverage, and handle class imbalance. A stochastic gradient descent optimizer is used with an early stopping condition after 10 epochs with no validation improvement. Frozen pre-trained features $\mathbf{f}_t^{(i)}$ are cached to speed up training.

D. Accumulating Observations into Costmaps

While individual observations and predictions may contain visual ambiguity or occlusion, fusing these uncertain predictions over time makes *FITAM* more robust to these uncertainties. During deployment, inference is run on each image slice $\mathbf{I}_t^{(i)}$. This produces far-field predictions of distributions over cost classes $\mathbf{P}_t^{(i,j)}$ for each radial bin. These radial bins are projected over a Cartesian grid that stores mean cost $\hat{\mu}_{\text{cost}}$ and estimate variance σ^2_{cost} per cell. In order to accumulate observations of varying confidence over time we use a one-dimensional Kalman filter (similar to [28]) in each Cartesian cell. To use $\mathbf{P}_t^{(i,j)}$ for this Kalman update, we use the mean of the costs covered by the most likely cost class as the observed cost, and we set the observation variance to be proportional to the entropy of $\mathbf{P}_t^{(i,j)}$. Fixed process noise Q is added to all cells each map update step to capture errors in robot state estimation and bias estimates towards more recent observations. The Cartesian grid is then used to supplement an existing costmap. All Cartesian cells with σ^2_{cost} below a fixed threshold σ^2_{thresh} are used to overwrite *unobserved* cells in the existing costmap. The threshold prevents predictions with low confidence from entering the costmap. The combined map is then sent to the global planner.

V. Simulated Experiments

We first use simulation to benchmark *FITAM* at scale over long navigation tasks. Additionally, we examine the impact of the design decisions presented in this paper on the overall performance of the algorithm. To capture the impact of far-field predictions on navigation, we report results with respect to navigation without far-field assistance (*No FF*). This benchmark discards the far-field observations and does not incorporate them into the global map, but otherwise operates identically.

A. Simulated Experiment Setup

1) Environment: To capture real environment semantic distributions, we sample maps from the Chesapeake Bay Program (CBP) Land Use/Land Cover (LULC) dataset [29]. This dataset covers the 250,000 km^2 that make up the Chesapeake Bay watershed regional area (ranging from New York to Virginia, USA) at a meter-per-pixel resolution. Each pixel is assigned a land cover class (e.g., water, tree canopy). To use these maps for navigation, each land cover class is mapped to a human-chosen cost based on safe travel speeds on that terrain, ranging between $0.25\ \text{s}\,\text{m}^{-1}$ for road to $\infty\ \text{s}\,\text{m}^{-1}$ for water.

Obstacles (trees, bushes, rocks, fallen trees, buildings) are sampled based on the land cover class, and Fury [30] is used to render $360\,\text{deg}$ panoramas from different robot states in this environment. Each map is $5\text{km} \times 5\text{km}$. A sampled map and part of a panorama image can be seen in Figure 4. It is assumed that the robot can observe costs within a 25m radius of its current position that are not occluded by obstacles. These observations are termed local observations.

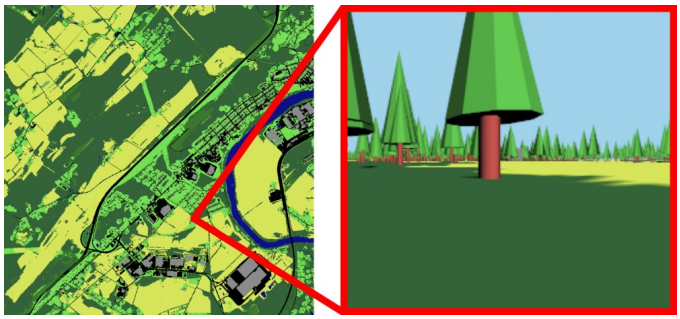

Fig. 4. An example $5\text{km} \times 5\text{km}$ semantic map sampled from [29] and a panorama slice rendered in this environment.

2) FITAM Configuration: Unless otherwise stated, the default *FITAM* configuration has three ($B = 3$) 25m radial bins spanning 25m to 100m and 32 radial bins of size $\Delta\theta_b = \frac{\pi}{16}$ radians, $\rho_{\text{class}} = 1$, $\rho_{\text{bin}} = 0.2$, $\rho_{\text{boot}} = 0.2$, $G = 3$, $M = 15$, $\sigma^2_{\text{thresh}} = 0.4$, $Q = 0.001$. The robot's max speed was $5\text{m}\,\text{s}^{-1}$. We found pre-trained feature extractors [26], [31] to degrade performance on simulated imagery, so we prepend an untrained ResNet-18 backbone to each model and train it jointly [31]. For planning we employ D*lite to search an 8-connected graph over the costmap [32].

Training data is generated by sampling a random sequence of valid poses in free-space in a map. The robot, starting from the first pose, navigates towards each sequential pose using *No FF*. Once a given time limit $T = t_{\text{max}}$ is reached, navigation stops and the partially observed map and egocentric images

are used to create datasets $\mathcal{D}^{\text{bal}}_{\text{ff,j}}$. Unless otherwise specified, $t_{\max}$ was ten hours. The dataset for the simulated trials was created from a map sampled from Baltimore County, MD.

3) Baseline and Additional Benchmarks: As a baseline, we implement [8], which is most similar to our work. [8] uses near-to-far self-supervised learning to predict traversability of distant terrain from images. However, [8] does not learn to predict the robot's expected traversal speed in a region of the map and instead uses geometry as a supervision signal to predict binary traversability of a distance-normalized image patch. This predicted traversability is then projected onto a ground plane to be placed in the robot's map. To fairly compare [8] to *FITAM*, we extend its prediction range from 40m to match *FITAM* at 100m. The same trajectory used to create *FITAM*'s dataset (see V-A2) was used to create a balanced dataset to train [8], which achieved similar classification accuracy on a withheld validation set as mentioned in the paper. No online adaptation of the model was employed during evaluations.

For simulated results, we additionally report the performance of two algorithms that are pseudo-upper bounds: *GT FF*, which receives ground truth far-field labels instead of using a learned model to predict bin speeds and *Perfect Vision*, which can perfectly observe the costmap out to 100m instead of 25m. Roughly, the difference between *FITAM* and *Ground Truth FF* captures the impact of the difficulty of predicting the labels on navigation, and the difference between *Ground Truth FF* and *Perfect Vision* captures the impact of *FITAM* discretization on navigation performance. Neither are true upper-bounds as better informed local decisions can be sub-optimal globally.

4) Evaluation: For evaluation, a map was sampled from the majority of the remaining counties after excluding Baltimore County in [29] as it was used for training data. One hundred start locations are sampled per map, each paired with a feasible sampled goal no less than one kilometer away to produce 18,400 evaluation planning problems.

During evaluation the agent iterates through a sense-plan-act loop, where it adds its local and *FITAM* observations to its costmap, plans a path in this updated costmap, then travels along this path. The agent travels until encountering a cell not observed by local observations, or a maximum of 20m along its path, whichever is shorter, to encourage sufficiently dense observations. A trial fails and is excluded if an approach does not reach the goal in 1000 iterations. Our primary metric of interest is the total cost accumulated while en route to the goal.

B. General Performance

Table I reports performance summaries of *FITAM*, the baseline, and the pseudo-upper bound algorithms on the 18,400 planning trials shown in Figure 5. All values are reported with the mean and 95% confidence intervals using the standard error of the mean. As shown in the table, *FITAM* reduces planning costs in $92.5 \pm 0.4\%$ of trials and on average reduces cost by $19.1 \pm 0.3\%$ compared to *No FF*, outperforming the baseline which improves $65.2 \pm 0.6\%$ of trials with an average cost reduction of $0.8 \pm 0.24\%$. This supports our hypothesis that *FITAM* is able to provide global planning guidance from images across a range of environments and planning problems.

	% Trials Imp ↑	Avg. % Diff ↑	% Failed ↓
Hadsell et al [8]	65.2 ± 0.6	0.8 ± 0.24	0.55
FITAM	92.5 ± 0.4	19.1 ± 0.3	0.74
GT FF	92.5 ± 0.4	19.2 ± 0.3	0.72
Perfect Vision	95.7 ± 0.3	24.7 ± 0.3	0.33

TABLE I. Performance compared to *No FF* across all 18,400 evaluation trials. % Trials Imp reports the percentage of trials improved with the approach (achieved lower cost) compared to *No FF*. Avg % Diff reports the average percent difference across all trials between the approach and *No FF*. % Failed reports the percentage of trails that failed. Positive values indicate reduction in cost. *No FF* failed 0.71% of trials.

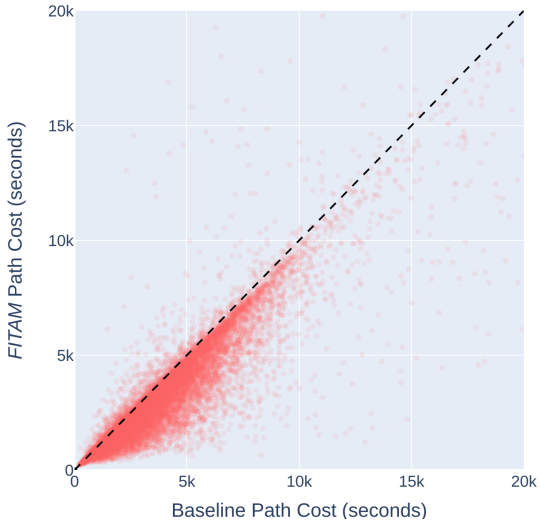

Fig. 5. Cost (measured in traversal time from start to goal) of *FITAM* (Y-axis) and baseline [8] (X-axis) in 18,400 simulated planning problems across 184 environments. Each point is a planning trial. The bias below the line $y = x$ indicates *FITAM* better learns to exploit far-field information for motion planning.

C. Ablations

To understand the impact of design decisions made for *FITAM*, we perturb parts of the architecture described in Section IV. These components and their alternatives are described below. For all ablations, the same evaluation configuration previously described is used, except only 10 planning problems are sampled from each map. Results are presented in Table II. All approaches failed in fewer than 1.1% of trials.

1) Dataset Collection Time: To understand the relationship between the amount of simulated data and performance, we take the full 10-hour trajectory collected for Section V-B and train a *FITAM* model on datasets created from subsets of the trajectory ranging from 10 minutes to 6 hours. We see in Table II-1 that *FITAM* improves navigation even with little data, but performance improves with more data.

2) Max Prediction Distance: We vary the maximum distance from the robot the range bins reach. One model is trained with eleven 25m range bins spanning a total distance from 25m to 300m. Trials are run while ignoring predictions from bins beyond the reported max distance (e.g., for a max distance of 250m, bins 10 and 11 are ignored). As shown in Table II-2 increasing range improves performance in the simulated setting, with diminishing returns by 200m.

3) Range Bin Fidelity: Without changing the total distance the range bins cover (25m to 100m), we vary the number (1,

1) Dataset Collection Time

Minutes	10 min	30 min	60 min	120 min	360 min
% Trials Imp	80.8 ± 1.8	82.7 ± 1.7	88.6 ± 1.5	91.6 ± 1.3	92.3 ± 1.2
Avg % Diff	7.4 ± 1.5	9.0 ± 1.6	15.3 ± 1.1	18.2 ± 1.0	18.9 ± 1.0

2) Max Prediction Distance

Max Dist (m)	100m	150m	200m	250m	300m
% Trials Imp	92.3 ± 1.2	94.1 ± 1.1	94.5 ± 1.0	94.2 ± 1.1	94.0 ± 1.1
Avg % Diff	18.7 ± 1.0	22.8 ± 1.0	24.6 ± 1.1	25.2 ± 1.1	25.4 ± 1.1

3) Range Bin Fidelity

# Rad Bins	1	2	4	8	10
% Trials Imp	87.3 ± 1.5	91.6 ± 1.3	92.7 ± 1.2	93.6 ± 1.1	93.9 ± 1.1
Avg % Diff	15.5 ± 1.0	18.5 ± 1.0	19.3 ± 1.0	20.1 ± 1.0	20.2 ± 1.0

4) Heading Bin Fidelity

# Ang Bins	4	8	16	32	64
% Trials Imp	80.4 ± 1.8	87.1 ± 1.5	92.1 ± 1.2	92.2 ± 1.2	92.5 ± 1.2
Avg % Diff	9.5 ± 1.1	14.1 ± 1.1	18.6 ± 1.0	19.0 ± 1.0	18.5 ± 1.0

5) Range and Heading Bin Fidelity

Bin Conf	1r4h	2r8h	4r16h	8r32h	10r64h
% Trials Imp	79.9 ± 1.8	86.1 ± 1.6	91.7 ± 1.3	93.4 ± 1.1	93.9 ± 1.1
Avg % Diff	6.7 ± 1.5	13.1 ± 1.1	18.9 ± 1.0	20.3 ± 1.0	19.9 ± 1.0

6) Number of Cost Classes

# Classes	2	3	4	5	6
% Trials Imp	45.1 ± 2.3	92.2 ± 1.2	89.6 ± 1.4	89.5 ± 1.4	90.4 ± 1.4
Avg % Diff	-1.1 ± 1.2	19.3 ± 1.0	14.8 ± 1.2	15.0 ± 1.1	18.0 ± 1.3

TABLE II. Results of ablation studies on 1840 planning trials. On these 1840 planning trials *Perfect Vision* achieved a % Trials Imp rate of $95.7 \pm 0.9\%$ and an Avg % Diff of $24.5 \pm 1.1\%$ and *Ground Truth FF* achieved a % Trials Imp rate of $93.1 \pm 1.2\%$ and an Avg % Diff of $19.1 \pm 1.0\%$

2, 4, 8, 10) of range bins. As the number of bins increases the environment is more finely discretized. As shown in Table II-3 after a noticeable jump between 1 and 2 range bins, performance only increases modestly with more radial bins.

4) Heading Bin Fidelity: Similar to Range Bin Fidelity, we vary the number of angular bins that the $360\,\mathrm{deg}$ around the robot is split into, using 4, 8, 16, 32, and 64 bins. Shown in Table II-4 the choice of angular discretization has a stronger impact on navigation performance compared to range binning, suggesting that a low number of heading bins seriously limits the usability of the predictions for navigation.

5) Range and Heading Bin Fidelity: Heading and range binning are varied jointly with the values used in the previous two fidelity ablations. In Table II-5 bin configurations are reported as XrYh indicating X radial bins and Y heading bins. Performance mirrors the heading fidelity ablation in trend, with a slight edge for higher bin count configurations.

6) Number of Cost Classes: We vary the number of cost classes the network can produce, increasing the fidelity at which *FITAM* can represent costs. All class configurations divide the space between $0\ \mathrm{m\,s^{-1}}$ and $5\ \mathrm{m\,s^{-1}}$ into a different number of (not necessarily uniform) bins, grouping together different subsets of the obstacles and terrain types present in the simulated maps. Results are shown in Table II-6. In this environment, 2 classes is not rich enough to guide motion planning effectively, while 3 performs well. Beyond 3 classes performance varies. This may reflect factors like learning complexity or how cost classes line up with observed costs.

VI. Physical Experiments

We demonstrate our approach on a Clearpath Robotics Warthog (shown in Figure 1), equipped with an OS1-64 lidar and FLIR Blackfly RGB forward-facing camera capturing images of size $H = 1080, W = 1440$. Data was collected during midday in the summer by teleoperating the robot for 12 minutes on one part of a road-forest junction as shown in Figure 6. The environment contained occluding obstacles like trees, fallen trees, and bushes. This data was used to train a model as described in Section IV. Image features were extracted with DinoV2 ViT-B [26]. *FITAM* was configured with two cost classes ($G = 2$, to capture road/forest) spanning $0 - 5\mathrm{m\,s^{-1}}$, four radial bins ($B = 4$) spanning $25\mathrm{m}$ to $100\mathrm{m}$, an angular bin width of $\Delta\theta_b = \frac{\pi}{20}\mathrm{rad}$, $\rho_{\text{class}} = \infty$, $\rho_{\text{bin}} = 0.2$, $\rho_{\text{boot}} = 0.2$, $M = 15$, $\sigma^2_{\text{thresh}} = 0.4$, and $Q = 0.001$.

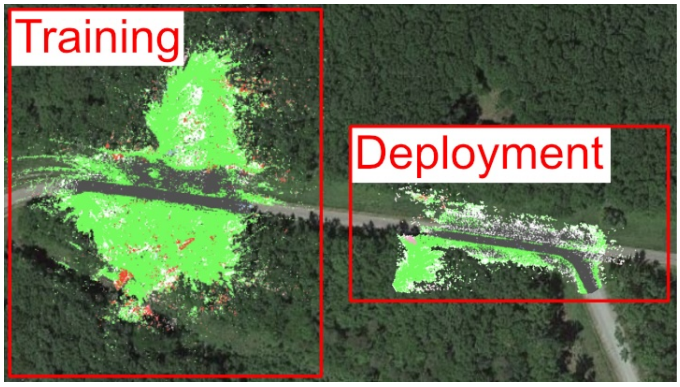

Fig. 6. Semantic costmaps $\mathbf{m}_l$ built from training data collected in the location shown on the left and (for visualization only) testing data on the right overlaid on a satellite image of the area.

On the physical platform we used the Kinodynamic Efficiently Adaptive State Lattice (KEASL) to produce global plans [33]. This planner optimizes the time of a feasible path while avoiding obstacles and obeying terrain speed limits. In our case, these terrain speed limits were the costs inferred by the *FITAM* model, combined with speed wells around obstacles to promote slower navigation when precision is required. We compare results against *No FF* that discards the far-field predictions but otherwise operates identically. The robot used a lidar-based obstacle classifier, marking all objects above $0.4\mathrm{m}$ as obstacles. SLAM was performed by OmniMapper [34] and local planning was handled by MPPI [35]. *FITAM* was run on a computer equipped with an Intel(R) Core(TM) i7-8700 12-core CPU, 32 gigabytes of RAM, and a Tesla T4 GPU. *FITAM* produced predictions at $1\mathrm{Hz}$. The trained model was deployed months later with start and goal locations shown in Figure 8. The environment had gone through significant visual change as shown in Figure 9. Three trials for each *FITAM* and the *No FF* benchmark were completed. When the platform collided with an obstacle or otherwise engaged in dangerous behavior, a teleoperator intervened, and safely completed the maneuver the robot was attempting.

A. Experimental Results

The GPS trajectory of each trial is shown in the center of Figure 8. In each of the *FITAM* trials, the robot picks out the road in the distance and global planning exploits it, bringing the robot to the goal more reliably (fewer interventions) and faster (total autonomy time), as reported in Figure 7 and Table III. When traveling on the road *FITAM* periodically detects the

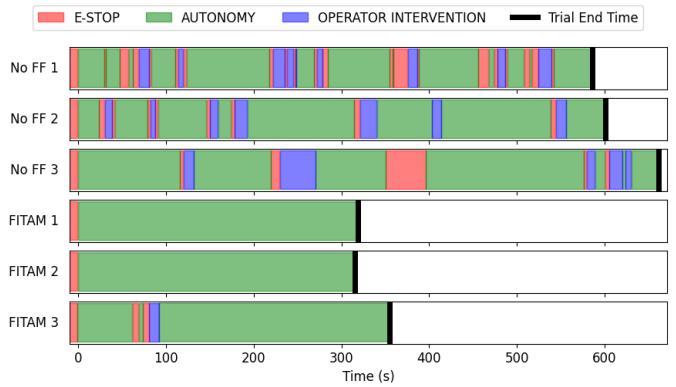

Fig. 7. Visual of robot status during trials. Autonomy (green) shows when the robot was under its own control, E-stop (red) when it was emergency stopped, and Manual (blue) when it was teleoperated during an intervention. *FITAM* significantly reduced the number of interventions during navigation.

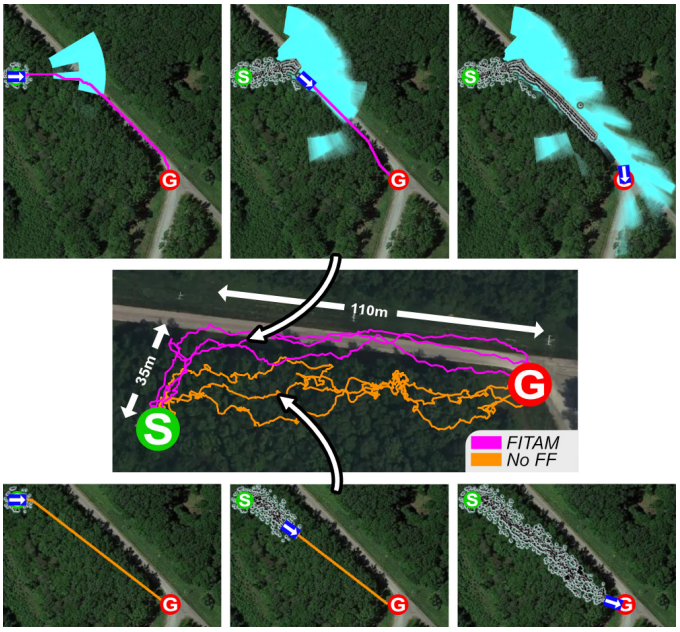

Fig. 8. Map evolution across a trial for *FITAM* and *No FF*. The center image shows the GPS position histories for all trials. The top (*FITAM*) and bottom (*No FF*) images show the global maps and global plans at the start (left), intermediate time (center) and end (right) of a trial. Black and grey regions display obstacles (trees and people operating the robot). In the *FITAM* maps shades of cyan show the accumulated far-field low cost predictions. The blue rectangle is the robot's position as it navigates between the start (green) and goal (red). As can be seen in the top left image, *FITAM* identifies the low cost road and clearing ranging 25m to 62m.

road in the furthest bin, adding it to the map up to 100m away. The *No FF* trials did not exploit the road and navigated directly towards the goal, bringing the robot through the rougher, forested terrain where frequent interventions were required to prevent collisions with trees and free the robot when it became stuck on logs or between trees.

B. Inference Performance Across Seasons

As highlighted by the far-field map accuracy in Figure 8, *FITAM* generalized well across the visual change shown in Figure 9 from summer (training) to fall (deployment). We

	Total Time (s)	Autonomy Time (s)	Interventions
No FF	615 (585, 600, 660)	464 (401, 483, 507)	6.3 (7, 7, 5)
FITAM	329 (318, 315, 354)	320 (318, 315, 329)	0.3 (0, 0, 1)

TABLE III. Performance metrics for real-world deployment: Values are displayed as Avg. (Trial 1, Trial 2, Trial 3). *FITAM* reduced the average autonomous time and total time (including interventions) to reach the goal by 31% and 46% respectively. One intervention occurred during *FITAM* trials compared to 19 during the *No FF* runs.

Fig. 9. *FITAM* was trained on data collected in summer and deployed in the fall. We also test classification performance across seasons (summer, fall, winter) as shown in Table IV.

observed similar map quality while running the same *FITAM* model on data collected in snowy winter conditions. To investigate seasonal generalization and the role pre-trained features play, we create five datasets: *train*/*val* from the training route in the summer shown in Figure 6 and *test summer*, *test fall*, and *test winter* from the deployment route in the respective seasons. On these datasets we train and evaluate *FITAM* models with three types of image features: DinoV2 ViT-B features (86.6M params) [26], ResNet-18 features pre-trained on ImageNet (11.2M params) [31], and ResNet-18 with no pre-trained weights trained jointly (No Pre). Fifteen models were trained with these features on the *train* dataset and class-weighted ROC AUC is reported on test datasets in Table IV. We use the same *FITAM* configuration as Section VI and use the datasets for the bin spanning 43m to 64m. We observe that the models using DinoV2 features generalize better to fall and winter datasets for predicting far-field costs.

C. Use of Ensembles

Ensembles were used in *FITAM* to reduce prediction confidence on out-of-distribution (OOD) data. To evaluate the ability of ensembles to reduce OOD confidence we trained an ensemble of $M = 15$ models with DINO features on the previous *train* dataset. We compare the ensemble confidence across all images in an *ood* dataset collected in an urban setting with the confidence of a single model ($M = 1$). As most of the images in the *ood* dataset contain situations not encountered in training (e.g., inside buildings, water), lower model confidence is desirable. We observed the ensemble reduced the average model confidence across the *ood* dataset from 88% to 80% on the binary classification task. The value M was chosen for the experiments above as most of the reduction in OOD uncertainty as a function of M was realized by $M = 15$.

	val	*summer test*	*fall test*	*winter test*
DinoV2	$1.0 \pm .000$	$.94 \pm .001$	$.63 \pm .010$	$.74 \pm .004$
ResNet18	$.99 \pm .000$	$.96 \pm .001$	$.58 \pm .001$	$.69 \pm .004$
No Pre	$.98 \pm .002$	$.97 \pm .001$	$.60 \pm .020$	$.67 \pm .036$

TABLE IV. Class-weighted ROC AUC for cost classification, comparing different image feature sets.

VII. Conclusions

In this work, we propose *FITAM* which learns to predict distant terrain costs using only images. *FITAM* trains on self-labeled data and integrates smoothly with existing navigation architectures that use costmaps. We demonstrate *FITAM*'s ability to outperform prior work at scale in a simulated setting, and inspect components of our algorithm through ablations. Finally, we deploy *FITAM* on a Clearpath Robotics Warthog and demonstrate its ability to detect distant and actionable low-cost regions which improved navigation performance against a baseline, even with significant visual change from training data. Possible directions for future work include methods that learn priors over environment structure to better leverage limited far-field information, or investigations into *FITAM*'s robustness to noisy self-supervised labels.

ACKNOWLEDGMENT

The authors acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing HPC resources that have contributed to research results.

References

[1] S. Goldberg, M. Maimone, and L. Matthies, "Stereo vision and rover navigation software for planetary exploration," in *Proceedings, IEEE Aerospace Conference*, vol. 5, 2002, pp. 5–5.

[2] D. Shah and S. Levine, "ViKiNG: Vision-Based Kilometer-Scale Navigation with Geographic Hints," in *Proceedings of Robotics: Science and Systems (RSS)*, 2022.

[3] B. Sofman, E. Lin, J. A. Bagnell, J. Cole, N. Vandapel, and A. Stentz, "Improving robot navigation through self-supervised online learning," *Journal of Field Robotics*, vol. 23, no. 11-12, pp. 1059–1075, 2006.

[4] 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.

[5] M. Bajracharya, B. Tang, A. Howard, M. Turmon, and L. Matthies, "Learning long-range terrain classification for autonomous navigation," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2008, pp. 4018–4024.

[6] M. J. Procopio, J. Mulligan, and G. Grudic, "Learning terrain segmentation with classifier ensembles for autonomous robot navigation in unstructured environments," *Journal of Field Robotics*, vol. 26, no. 2, pp. 145–175, 2009.

[7] O. Mayuku, B. W. Surgenor, and J. A. Marshall, "A self-supervised near-to-far approach for terrain-adaptive off-road autonomous driving," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 14 054–14 060.

[8] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, J. Han, B. Flepp, U. Muller, and Y. LeCun, *Online learning for offroad robots: Using spatial label propagation to learn long-range traversability*, 2008, pp. 17–23.

[9] Z. Li, Z. Chen, X. Liu, and J. Jiang, "Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation," *Machine Intelligence Research*, vol. 20, pp. 837–854, 2023.

[10] V.-C. Miclea and S. Nedevschi, "Monocular depth estimation with improved long-range accuracy for uav environment perception," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, 2022.

[11] P. Borges, T. Peynot, S. Liang, B. Arain, M. Wildie, M. Minareci, S. Lichman, G. Samvedi, I. Sa, N. Hudson, *et al.*, "A survey on terrain traversability analysis for autonomous ground vehicles: Methods, sensors, and challenges," *Field Robot*, vol. 2, pp. 1567–1627, 2022.

[12] D. Silver, J. A. Bagnell, and A. Stentz, "Learning from demonstration for autonomous navigation in complex unstructured terrain," *The International Journal of Robotics Research*, vol. 29, pp. 1565–1592, 2010.

[13] Z. Zhu, N. Li, R. Sun, D. Xu, and H. Zhao, "Off-road autonomous vehicles traversability analysis and trajectory planning based on deep inverse reinforcement learning," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 971–977.

[14] X. Meng, N. Hatch, A. Lambert, A. Li, N. Wagener, M. Schmittle, J. Lee, W. Yuan, Z. Chen, S. Deng, G. Okopal, D. Fox, B. Boots, and A. Shaban, "TerrainNet: Visual Modeling of Complex Terrain for High-speed, Off-road Navigation," in *Proceedings of Robotics: Science and Systems (RSS)*, Daegu, Republic of Korea, 2023.

[15] M. G. Castro, S. Triest, W. Wang, J. M. Gregory, F. Sanchez, J. G. Rogers, and S. Scherer, "How does it feel? self-supervised costmap learning for off-road vehicle traversability," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 931–938.

[16] X. Cai, M. Everett, J. Fink, and J. P. How, "Risk-aware off-road navigation via a learned speed distribution map," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2931–2937.

[17] J. Sock, J. Kim, J. Min, and K. Kwak, "Probabilistic traversability map generation using 3d-lidar and camera," in *International Conference on Robotics and Automation (ICRA)*, 2016, pp. 5631–5637.

[18] H. Lu, L. Jiang, and A. Zell, "Long range traversable region detection based on superpixels clustering for mobile robots," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 546–552.

[19] A. Yang, W. Li, and Y. Hu, "F3dmp: Foresighted 3d motion planning of mobile robots in wild environments," in *International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan, 2024.

[20] M. Ryll, J. Ware, J. Carter, and N. Roy, "Semantic trajectory planning for long-distant unmanned aerial vehicle navigation in urban environments," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 1551–1558.

[21] C. Bradley, A. Pacheck, G. J. Stein, S. Castro, H. Kress-Gazit, and N. Roy, "Learning and planning for temporally extended tasks in unknown environments," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4830–4836.

[22] A. Krebs, C. Pradalier, and R. Siegwart, "Adaptive rover behavior based on online empirical evaluation: Rover–terrain interaction and near-to-far learning," *Journal of Field Robotics.*, vol. 27, no. 2, 2010.

[23] E. Chen, C. Ho, M. Maulimov, C. Wang, and S. Scherer, "Learning-on-the-drive: Self-supervised adaptation of visual offroad traversability models," *arXiv preprint arXiv:2306.15226*, 2023.

[24] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Proceedings of Robotics: Science and Systems (RSS)*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[25] P. Chao, C.-Y. Kao, Y.-S. Ruan, C.-H. Huang, and Y.-L. Lin, "Hardnet: A low memory traffic network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3552–3561.

[26] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "Dinov2: Learning robust visual features without supervision," *Transactions on Machine Learning Research Journal*, pp. 1–31, 2024.

[27] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[28] S. Triest, D. D. Fan, S. Scherer, and A.-A. Agha-Mohammadi, "Unrealnet: Learning uncertainty-aware navigation features from high-fidelity scans of real environments," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[29] C. B. Program, "Chesapeake bay land use and land cover (lulc) database 2022 edition," *Data Release*, 2022.

[30] E. Garyfallidis, S. Koudoro, J. Guaje, M.-A. Côté, S. Biswas, D. Reagan, N. Anousheh, F. Silva, G. Fox, and F. Contributors, "Fury: advanced scientific visualization," *Journal of Open Source Software*, vol. 6, 2021.

[31] 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 (CVPR)*, 2016, pp. 770–778.

[32] S. Koenig and M. Likhachev, "D* lite," in *Eighteenth National Conference on Artificial Intelligence*, 2002, pp. 476–483.

[33] E. R. Damm, J. M. Gregory, E. S. Lancaster, F. A. Sanchez, D. M. Sahu, and T. M. Howard, "Terrain-aware kinodynamic planning with efficiently adaptive state lattices for mobile robot navigation in off-road environments," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 9918–9925.

[34] A. J. B. Trevor, J. G. Rogers, and H. I. Christensen, "Omnimapper: A modular multimodal mapping framework," in *International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1983–1990.

[35] 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 *International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1714–1721.