

Learning When to Jump for Off-road Navigation

Zhipeng Zhao, Taimeng Fu, Shaoshu Su, Qiwei Du,
Ehsan Tarkesh Esfahani, Karthik Dantu, Souma Chowdhury, and Chen Wang
University at Buffalo, NY 14260, USA

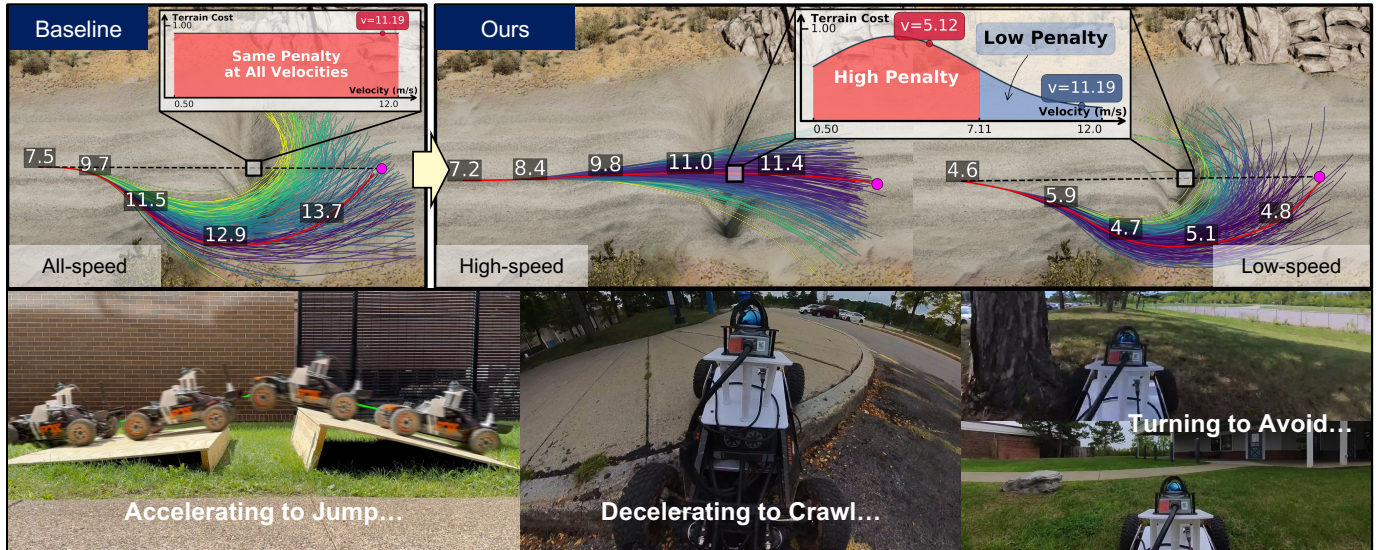


Fig. 1: Challenging terrains often require motion strategies that span from aggressive maneuvers to cautious crawling. Our Motion-Aware Traversability (MAT) model enables the vehicle to decide when to accelerate to jump over a ditch, slow down to crawl over curbs, and steer to avoid obstacles. In contrast, baseline methods assign uniformly high costs to features like ditches, causing the vehicle to simply avoid them rather than selecting motion strategies that could traverse them more efficiently.

Abstract—Low speed does not always guarantee safety in off-road driving. For instance, crossing a ditch may be risky at a low speed due to the risk of getting stuck, yet safe at a higher speed with a controlled, accelerated jump. Achieving such behavior requires path planning that explicitly models complex motion dynamics, whereas existing methods often neglect this aspect and plan solely based on positions or a fixed velocity. To address this gap, we introduce Motion-aware Traversability (MAT) representation to explicitly model terrain cost conditioned on actual robot motion. Instead of assigning a single scalar score for traversability, MAT models each terrain region as a truncated Gaussian function of velocity. During online planning, we decompose the terrain cost computation into two stages: (1) predict terrain-dependent Gaussian parameters from perception in a single forward pass, (2) efficiently update terrain costs for new velocities inferred from current dynamics by evaluating these functions without repeated inference. We develop a system that integrates MAT to enable agile off-road navigation and evaluate it in both simulated and real-world environments with various obstacles. Results show that MAT achieves real-time efficiency and enhances the performance of off-road navigation, reducing path detours by 75% while maintaining safety across challenging terrains. Code is available at <https://github.com/sair-lab/mat>.

I. INTRODUCTION

Off-road navigation presents unique challenges for autonomous robots [2, 3, 35], where safety and efficiency depend not only on terrain geometry but also on the robot’s motion dynamics. For example, crossing a ditch at a moderate speed may lead to getting stuck at the bottom, but a higher, controlled

speed can generate sufficient momentum to jump over it safely. Conversely, navigating bumpy roads and ruts may only be safe at low speeds to prevent instability. Such situations highlight that traversability is not a static property of terrain appearance, as it must also account for how the robot moves.

Most existing traversability estimation methods overlook this coupling between terrain and motion dynamics. They typically only infer position-dependent cost maps from terrain appearance and assign static scalar values to each location, independent of the robot’s intended motion. These methods include analytical approaches based on geometric features [18, 10] and learning-based models trained on various supervisions such as semantic labels [20], terrain elevation [6], past vehicle-terrain interactions [27, 29, 7], and expert trajectories [28, 15, 32]. While effective for avoiding obstacles or following smooth trails, these methods share a fundamental limitation: they cannot distinguish situations where the same terrain can be safe or unsafe depending on the robot’s speed or motion profile. In reality, traversability is motion-dependent, and ignoring it yields conservative or suboptimal plans, which could be problematic in time-critical missions [22].

Recent efforts have attempted to consider robot motion in terrain cost computation, but significant limitations remain. Analytical formulations struggle to generalize, as precise physical models that capture velocity effects across varied terrains are difficult to derive. For example, Han et al. [14] estimate

vehicle orientations from terrain elevations under a constant ground-contact assumption and combine them with velocities to compute traversability constraints via simplified physics. However, this assumption excludes agile maneuvers such as jumps, and the terrain geometry is reduced to a slope-based feature, limiting its applicability to complex terrains. From a learning perspective, HDIF [5] jointly inputs the terrain patch and the robot’s velocity into the neural network. However, since this requires repeated neural inferences for every velocity change, real-time cost updates during optimization are computationally infeasible. In practice, HDIF assumes a fixed velocity, failing to incorporate actual motion into traversability cost. Furthermore, learning the traversability-velocity relationship is difficult without providing labels covering the full velocity spectrum, and therefore, its predictions tend to be dominated by visual features. These limitations motivate a traversability representation that can efficiently condition terrain cost on both terrain perception and actual robot motion for real-time use.

To address this gap, we propose a Motion-aware Traversability (MAT) representation that explicitly conditions terrain cost on the robot’s motion dynamics. As illustrated in Fig. 1, each pixel is represented as a function of velocity rather than a scalar score. The function is modeled as a truncated Gaussian over a fixed velocity range, parameterized by terrain-dependent attributes: peak difficulty, the velocity at which it occurs, and sensitivity to deviations. These compact parameters capture richer terrain characteristics than a single score. Moreover, because these Gaussian functions can be reliably estimated from only a few trials, it enables the model to extrapolate traversability to velocities never observed during training, providing wider coverage over untested motion conditions. This property further strengthens MAT’s ability to generalize to out-of-distribution conditions.

We further develop an off-road navigation system by integrating MAT with a Model Predictive Path Integral (MPPI) [37] planner, enabling real-time adaptation of terrain costs to the motion inferred under system dynamics. The computation is decomposed into two stages: (1) a single neural inference predicts Gaussian parameters for the entire map, and (2) as the motion changes, new terrain costs can be rapidly inferred by calling these functions without re-running the perception network. This allows the robot to reason about when a dynamic maneuver, such as a jump, is safer and faster, ultimately planning more efficient and robust off-road trajectories.

Our main contributions can be summarized as:

- We introduce a motion-aware traversability (MAT) representation that conditions terrain cost on both perception and motion, enabling robust and agile off-road planning.
- Based on MAT, we develop an efficient off-road navigation system that knows when to jump, enabling real-time motion-conditioned terrain cost updates and optimization.
- We validate the system in diverse simulated (BeamNG) and real-world environments with various obstacles, demonstrating its real-time efficiency and reduced traversal distance, time, and energy over the baseline.

II. RELATED WORKS

Traversability quantifies the difficulty of moving through a region and serves as the terrain cost in off-road navigation. Most methods infer a value per region solely from perception, resulting in position-only traversability. Recent studies consider how motion affects difficulty and incorporate this into the traversability, which we term motion-enhanced traversability.

A. Position-only Traversability

Classical analytic approaches compute local geometric features such as roughness, slope, step height, and curvature from LiDAR scans and derive traversability from these statistics [18, 10]. However, LiDAR sparsity at high speeds and the lack of semantic context limit their applicability. Learning-based methods thus predict traversability from RGB/height images. These methods differ mainly in their supervision signals and learning strategies. Some construct labels offline from point clouds and semantic information, then train image-based predictors [20, 6]. For instance, TerrainNet [20] maps image inputs to a multi-layer representation of elevation and semantic probabilities, which are then converted into terrain costs.

To avoid manual annotation and hard-coded mapping from semantic to cost, self-supervised learning-based methods use vehicle-terrain interaction signals measured by onboard sensors (e.g., IMU, force/torque) as supervision. They derive labels such as vertical forces [36, 27, 29] or discrepancies between expected and actual pose/velocity [7], enabling automatic label generation from driving data. Another direction leverages expert demonstrations as supervision. In [28], previously driven regions are labeled as traversable, and contrastive learning is used to generalize to visually similar but unlabeled regions. V-STRONG [15] extends it by incorporating SAM-based [17] segmentation masks in contrastive learning. In [32], inverse reinforcement learning (IRL) is employed to learn a cost function, under which expert trajectories are optimal.

These methods produce position-only traversability maps that ignore motion effect. They can guide a robot toward flat regions and away from obstacles, but cannot tell how motion should adapt to specific terrain. For example, they may assign a low cost to a mildly bumpy road, but can not suggest a reduced speed, or mark a ditch as untraversable though it could be crossed via a jump. This limitation leads to suboptimal plans.

B. Motion-enhanced Traversability

A physics-based method [14] estimates vehicle orientations from the elevation map and combines them with vehicle velocities to calculate traversability constraints, aiming to prevent rollover and vehicle-ground separation. However, its use of orientation-only terrain features fails to capture richer terrain geometry, and the assumption that the vehicle always adheres to the ground ignores agile strategies such as controlled jumps.

Learning-based methods capture the motion effect directly from data. HDIF [5] conditions its prediction on velocity by fusing Fourier-encoded speed signals with visual features. Yet any change in velocity requires a new network inference, making real-time optimization computationally infeasible. As

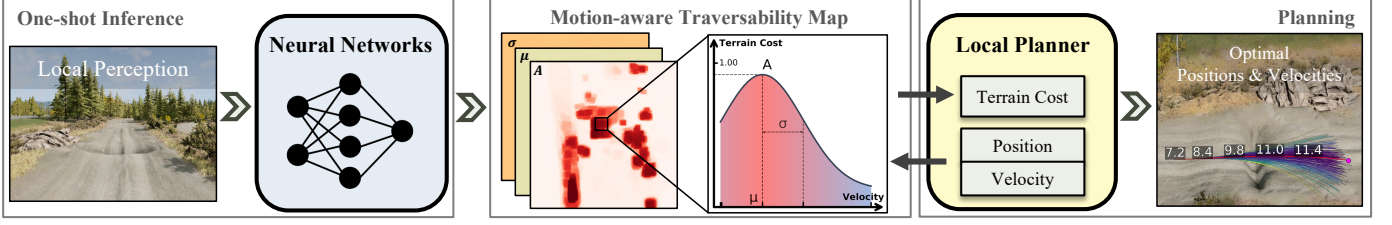


Fig. 2: The Architecture of the proposed pipeline. Given a local perception input (a LiDAR-based height map in practice), the neural network performs one-shot inference to generate the Motion-aware Traversability (MAT) map, where each cell encodes a Gaussian function of velocity. The local planner (e.g., MPPI) then samples trajectories based on vehicle dynamics, continuously querying the MAT map for terrain costs with position-velocity pairs to optimize motion in real time.

as a result, HDIF uses a fixed velocity during planning. Moreover, learning a robust traversability-velocity relationship is challenging without dense labels across speeds, and end-to-end predictions often become dominated by visual features. Alternatively, Velociraptor [31] and Salon [30] learn speed maps [39, 40] as auxiliary cost. They encourage adherence to expert velocities, such as high speeds on flat trails and lower speeds in vegetation, but only for regions that are deemed traversable by the traversability map. Since traversability remains motion-independent, terrains requiring specific motion strategies, such as jumping a ditch, are still penalized as untraversable. Besides, it fails to account for terrain-specific sensitivity, where a wide range of speeds might be safe.

Beyond traversability estimation, a line of work advances agile motion by extending vehicle kinodynamics [19, 9, 8, 22, 21, 42, 12]. PHLI [22] studies jumping behavior through in-air attitude control. It enables reorientation during airtime for safe landing and highlights the potential of aerial maneuvers for ground vehicles, but does not reason about when or where jumping is feasible, nor does it adapt to terrain geometry.

To handle complex off-road scenarios, we need a representation that conditions terrain cost on both perception and robot motion, while allowing efficient and reliable online evaluation.

III. APPROACH

A. Overview

A key challenge in off-road navigation is that traversability does not simply increase or decrease with speed: depending on the terrain, a moderate velocity can be hazardous while higher or lower speeds may ensure safety. To identify this property, a robot must model the traversability-velocity relationship and can extrapolate to unobserved velocities. Real datasets rarely contain dense velocity coverage, so directly regressing traversability from terrain-velocity input tends to fail on out-of-distribution velocities. To address this, we assume that traversability is a truncated Gaussian-shaped function of velocity. Crucially, since robot velocity is bounded by physical limits, this function is defined over a fixed range, allowing it to capture diverse terrain profiles including non-monotonic and monotonic risks, as illustrated in Fig. 3. This assumption is also motivated by its ability to yield smooth, physically plausible extrapolations from only a few samples, and its empirical effectiveness across diverse off-road scenarios. With

this assumption, we introduce the Motion-Aware Traversability (MAT) map that encodes this function at each terrain location.

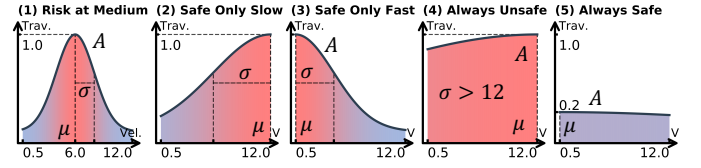


Fig. 3: Representative Gaussian-modeled profiles for common terrains: (1) long trenches, (2) rocky terrain/bumps, (3) narrow ditches/gaps, (4) trees/boulders, (5) smooth dirt trails.

The following sections describe: (1) how the MAT map is learned in a self-supervised manner from vehicle-terrain interaction, and (2) how a complete navigation system is built around the MAT map to enable real-time motion optimization.

B. Motion-aware Traversability Learning

The core idea of MAT is to predict *terrain-dependent Gaussian parameters* from perception rather than directly regressing the final traversability cost. Our neural model outputs a three-channel map where each position $\mathbf{x}$ is represented by: peak penalty $A(\mathbf{x})$, the velocity at which peak penalty occurs $\mu(\mathbf{x})$, and the sensitivity to velocity deviations $\sigma(\mathbf{x})$. Together, these parameters define a *velocity-conditioned terrain cost function* for each $\mathbf{x}$. This formulation captures more terrain properties, such as worst-case speed and sensitivity to deviations, instead of a single score that assumes uniform cost across speeds. Besides, by learning function parameters rather than final cost under different velocities, MAT enables efficient evaluation of terrain cost at arbitrary velocities without extra neural inference, and provides reliable extrapolation to unseen velocities.

Concretely, given a height map $\mathbf{H} \in \mathbb{R}^{H \times W}$, we aim to learn a network g_θ that predicts a MAT map $\hat{\Psi} \in \mathbb{R}^{H \times W \times 3}$. For each position $\mathbf{x}$, the MAT map stores a vector of Gaussian parameters $\psi(\mathbf{x}) = [A(\mathbf{x}), \mu(\mathbf{x}), \sigma(\mathbf{x})]^\top$, which jointly define a terrain traversability cost function over velocity:

$$T(\mathbf{x}, v) \triangleq T(v; \psi(\mathbf{x})) = A(\mathbf{x}) \exp\left(-\frac{(v - \mu(\mathbf{x}))^2}{2\sigma(\mathbf{x})^2}\right). \quad (1)$$

To learn g_θ , we first construct self-supervised labels Ψ from repeated vehicle-terrain interactions, and then train an image-to-image network with a masked regression objective.

1) *Self-supervised Label Construction*: To create the MAT map label in the form of Gaussian parameters, we must observe how traversability changes when the same terrain is traversed at different velocities. We therefore execute R runs over the same terrain with commanded speeds spanning $[v_{\min}, v_{\max}]$. Each trajectory is resampled to a common set of positions $\{\mathbf{p}_i\}_{i=0}^M$ [34, 41]. At each $\mathbf{p}_i$, we record the velocity $v_{r,i}$ and vehicle response in each run r , from which we derive the traversability cost $t_{r,i}$. Given a small set of velocity-traversability samples $\mathcal{S}_i = \{(v_{r,i}, t_{r,i})\}_{r=1}^R$, we estimate the Gaussian parameters $\psi(\mathbf{p}_i)$ by fitting the model in (1):

$$\psi(\mathbf{p}_i) = \arg \min_{\psi} \sum_{r=1}^R \left[t_{r,i} - T(v_{r,i}; \psi) \right]^2. \quad (2)$$

Here, sparse velocity observations suffice to recover a reliable traversability function over full velocities and provide effective supervision to the network via estimated parameters.

To quantify traversability cost $t_{r,i}$ at position $\mathbf{p}_i$ and velocity $v_{r,i}$, we derive it from vehicle-terrain interaction data rather than manual annotation. The intuition is that true traversability can only be revealed through real experience: visual appearance alone is insufficient to determine whether a rough-looking terrain is actually traversable. We define traversability by jointly considering *safety* and *effort*, which are detailed below.

a) *Safety (Rollover Risk)*: To capture the rollover risk under particular terrain-velocity combinations, we quantify it by how vehicle instability is amplified, rather than simply penalizing terrain unevenness as in prior methods [7, 12, 20]. A terrain-velocity pair is considered traversable if the vehicle either adheres closely to the terrain geometry or maintains smoother roll and pitch dynamics (e.g., a stable airborne transition). Conversely, it is deemed risky when vehicle roll or pitch angles are significantly amplified relative to the reference static orientation. We estimate vehicle static orientation (roll, pitch) from wheel contact elevations using a quasi-static model similar to [7]. Safe orientation bounds $\underline{\theta}$ and $\bar{\theta}$ are defined from the minimum and maximum static orientation angles, and the rollover risk is measured as the deviation of the vehicle's actual orientation $\theta_{r,i}$ from these bounds:

$$t_{r,i}^{\text{roll-raw}} = \max(0, \|\theta_{r,i} - \bar{\theta}\|_{\infty}, \|\underline{\theta} - \theta_{r,i}\|_{\infty}),$$

where $\|\cdot\|_{\infty}$ denotes the maximum deviation across roll and pitch. To account for delayed rollover effects (e.g., post-jump instability), we aggregate future instantaneous risks using exponential discounted factors $\gamma \in (0, 1)$. The accumulated value is then normalized to $[0, 1]$, with deviations of 90° or greater mapped to 1, yielding the final safety metric $t_{r,i}^{\text{roll}}$.

b) *Effort (Time and Energy)*: A region may be traversable with negligible rollover risk but require varying effort at different speeds. For example, a ditch can be crossed either slowly or via a high-speed jump, but a low speed takes longer and can consume more energy. To quantify this effort, we measure: (i) the travel time required to reach the goal, and (ii) the control energy, computed as the time integral of the throttle command from $\mathbf{p}_i$ to the goal. Since absolute values

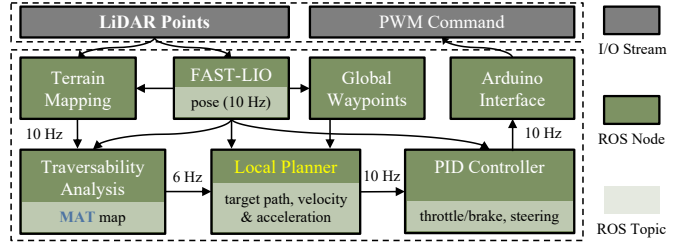


Fig. 4: Navigation system diagram showing data flow from LiDAR sensing to actuator commands, with ROS nodes performing perception, MAT estimation, planning, and control.

depend on the distance to the goal, we normalize each quantity across all runs at the same $\mathbf{p}_i$. This yields effort terms $t_{r,i}^{\text{time}}$ and $t_{r,i}^{\text{energy}}$, which are directly comparable across velocities. The resulting self-supervised traversability score at position $\mathbf{p}_i$ under velocity $v_{r,i}$ is defined as:

$$t_{r,i} = w_r t_{r,i}^{\text{roll}} + w_t t_{r,i}^{\text{time}} + w_e t_{r,i}^{\text{energy}}, \quad (3)$$

with $w_r = 0.8$, $w_t = 0.1$, $w_e = 0.1$ in our implementation.

After obtaining $\psi(\mathbf{p}_i)$ at $\mathbf{p}_i$, we assign it to the corresponding pixel and its surrounding patch on the height map. Repeating this process over all positions yields a ground-truth MAT map $\Psi \in \mathbb{R}^{H \times W \times 3}$ for $\mathbf{H}$. Since the label cannot cover the entire $\mathbf{H}$, we also generate a binary mask $\mathbf{M} \in \mathbb{R}^{H \times W}$ that indicates labeled pixels to enable image-to-image training.

2) *Neural Network*: To capture both local and global terrain geometry from $\mathbf{H}$, we utilize a U-Net-based [24] neural network as g_θ . The encoder extracts multi-scale geometric cues, and the decoder upsamples them to the original resolution to produce per-pixel outputs. To ensure valid physical ranges, we apply per-channel sigmoid activations followed by affine scaling. The network is trained using a masked mean-squared error loss computed over labeled pixels:

$$\mathcal{L}(\theta) = \frac{1}{\sum_{\mathbf{p}} \mathbf{M}(\mathbf{p}) + \varepsilon} \sum_{\mathbf{p}} \mathbf{M}(\mathbf{p}) \|g_\theta(\mathbf{H})(\mathbf{p}) - \Psi(\mathbf{p})\|_2^2. \quad (4)$$

Random spatial augmentations are applied during training to improve robustness. After training, the neural model learns to associate geometric terrain patterns with Gaussian parameters that characterize how traversability varies with speed. The resulting MAT map forms the basis for efficient, velocity-aware terrain cost evaluation during navigation.

C. Motion-aware Navigation System

We develop a complete navigation system that integrates MAT with perception, planning, and control modules for agile off-road driving. As illustrated in Fig. 4, the system ensures real-time motion planning when incorporating the MAT map, making motion-conditioned terrain cost optimization feasible on an edge device. The system is implemented in ROS [23], with each component running as an independent node. It receives LiDAR packets and outputs Pulse-Width Modulation (PWM) commands for throttle and steering.

The perception begins with FAST-LIO [38], providing LiDAR-inertial odometry at 10 Hz. A terrain mapping node,

implemented following [4], fuses recent registered scans to maintain a dense local point cloud. Our traversability analysis node then converts this local map into a 2D elevation map $\mathbf{H}$ and predicts the three-channel MAT map in a single inference.

The planning pipeline is initiated by the MAT map and the goal location $\mathbf{g}$ provided by a global waypoint node. Unlike prior systems that optimize only positions to minimize terrain cost, we jointly optimize both positions and velocities to produce terrain-adaptive motion strategies. This is enabled by integrating MAT such that standard local planners can evaluate terrain cost analytically for any new velocity. We demonstrate this using an MPPI planner, widely adopted in off-road navigation, as shown in Algorithm 1. At each iteration, N control sequences $\{\mathbf{u}_{0:T-1}^{(n)}\}_{n=1}^N$ are sampled around a nominal sequence and propagated through the vehicle dynamics:

$$\dot{\mathbf{x}} = v [\cos \psi \ \sin \psi], \quad \dot{\psi} = v\kappa, \quad \dot{v} = a, \quad \dot{\kappa} = u_\kappa, \quad (5)$$

where the control inputs $\mathbf{u} = [a, u_\kappa]$ are acceleration and curvature rate, and the state $\mathbf{s} = [\mathbf{x}, \psi, v, \kappa]$ includes position, heading, speed, curvature. Each trajectory is evaluated using:

$$\mathbb{J} = \sum_{t=0}^{T-1} [\mathbf{u}_t^\top \mathbf{R} \mathbf{u}_t + T(\mathbf{x}_t, v_t) + c_{\text{aux}}(\mathbf{s}_t, \mathbf{g})] + \phi(\mathbf{s}_T, \mathbf{g}), \quad (6)$$

where $\mathbf{R}$ penalizes control effort, $c_{\text{aux}}(\cdot)$ handles goal tracking and stability, and $\phi(\cdot)$ is the terminal cost. The terrain cost $T(\mathbf{x}_t, v_t)$ is analytically evaluated with Gaussian parameters $\psi(\mathbf{x}_t) = [A(\mathbf{x}_t), \mu(\mathbf{x}_t), \sigma(\mathbf{x}_t)]^\top$ available for $\mathbf{x}_t$ as in (1). Because $T(\mathbf{x}_t, v_t)$ is a closed-form function of velocity, it can be rapidly updated for a new velocity sampled during the iterative optimization process. Given that MAT provides the parameters for the entire local environment, the planner can efficiently evaluate any $(\mathbf{x}_t, v_t)$ pairs sampled across all iterations. After K iterations, the optimized path $\mathbf{x}_{0:T-1}^*$ with velocity and control profile is obtained, specifying both where to go and how aggressively to travel.

A low-level PID controller executes the plan: a look-ahead point on the optimized path determines the steering command, while the desired velocity and acceleration yield the throttle/brake command. An interface node translates these into PWM values and sends them over the serial port.

Conventional off-road navigation pipelines rely on position-only terrain cost $T(\mathbf{x}_t)$, which ignore the coupling between traversability and velocity and tend to produce suboptimal behaviors. In contrast, MAT provides a velocity-conditioned terrain cost $T(\mathbf{x}_t, v_t)$ that enables the planner to reason about how speed affects risk. The key contribution of our system is that it enables this richer, motion-aware traversability representation without increasing computational cost, ensuring real-time performance on resource-constrained platforms.

IV. EXPERIMENTS

A. Implementation Details

1) *Platforms*: We use both simulation and real-world setups for training and evaluation. Simulation is performed in BeamNG.tech [1], which provides high-fidelity soft-body

Algorithm 1 MPPI with Motion-aware Traversability (MAT)

```

1: Input: initial state  $\mathbf{s}_0$ ,  $\mathbf{u}_{0:T-1}$ , height map  $\mathbf{H}$ , goal  $\mathbf{g}$ 
2: Predict the MAT map  $(A, \mu, \sigma) \leftarrow g_\theta(\mathbf{H})$ 
3: for iteration  $k = 1$  to  $K$  do
4:   for each sampled control perturbation  $\delta \mathbf{u}_{0:T-1}^{(n)}$  do
5:     for  $t = 1$  to  $T - 1$  do
6:        $\mathbf{s}_t^{(n)} \leftarrow F(\mathbf{s}_{t-1}^{(n)}, \mathbf{u}_{t-1}^{(n)})$ ,  $(\mathbf{x}_t^{(n)}, v_t^{(n)}) \leftarrow \mathbf{s}_t^{(n)}$ 
7:       Accumulate cost  $T(\mathbf{x}_t^{(n)}, v_t^{(n)})$  and other terms
8:     end for
9:     Compute trajectory cost  $\mathbb{J}^{(n)}$ 
10:   end for
11:   Update  $\mathbf{u}_{0:T-1}$  via MPPI weighted averaging
12: end for
13: Output: optimal control  $\mathbf{u}_0^*$ 

```

physics for accurate vehicle dynamics and tire-terrain modeling. Its World Editor allows custom terrain geometry, enabling data collection for aggressive maneuvers that would be too costly in the real world. The real-robot platform is a Losi 1/5-scale *Racer* car equipped with a MID-360 Livox, an NVIDIA Jetson Nano, and an Arduino. The setup supports on-board sensing, computation, and execution in real-world navigation.

2) *Configurations*: The neural network is a lightweight U-Net with four encoder-decoder stages and channel widths of [16, 32, 64, 128]. Its input is a LiDAR-based height map generated on a $0.1\text{m} \times 0.1\text{m}$ grid within a 40 m sensing range. This range balances perceptual coverage with the foresight required for high-speed traversal. The model is trained in simulation using Adam [16] and fine-tuned on real-world data with a reduced learning rate. To avoid significant hardware damage, real data is collected in a conservative velocity range of 0.5 to 6.5 m/s. The MPPI is implemented within the MPPI-Generic [33] framework, a C++/CUDA header-only library for conducting stochastic optimal control. We implement the dynamics model and our traversability cost function that conditions the terrain cost on the sampled motion.

3) *Baselines*: We use a Position-Only Traversability (PO-Trav) that shares the same configuration as MAT but replaces the motion-aware terrain cost with a position-only formulation. This design provides a fair comparison that focuses on the contribution of our motion-aware representation by avoiding inconsistencies from different traversability definitions. Meanwhile, it captures the common essence of prior methods [7, 27, 29], and directly integrating these works is often hindered by the lack of full implementation releases [13] or the difference in target robot types (e.g., legged robots [11]). For comparison within the Motion-enhanced Traversability category defined in Section II, we include AnyNav [12] and PhysORD [42] in the simulation. Both methods go beyond traversability estimation by improving off-road vehicle dynamics models for navigation and provide simulation implementations, while other motion-enhanced approaches lack released code for direct comparison.

4) *Metrics*: To quantitatively assess off-road navigation, we use three metrics. Detour distance measures the path length relative to the straight-line distance between start and goal. While the values are not directly comparable across different

TABLE I: Quantitative comparison between MAT and the baselines in the obstacle traversal task. For each metric, the average value (**Avg.**) across three velocity levels (induced by T_1 , T_2 , T_3) is reported. In each comparison, the better result is in bold.

Terrain Geometry	Method	Detour Distance (m) ↓				Traversal Time (s) ↓				Energy Consumption ↓			
		T_1	T_2	T_3	Avg.	T_1	T_2	T_3	Avg.	T_1	T_2	T_3	Avg.
Short Ditch	PO-Trav	2.43	2.72	2.54	2.56	18.49	10.89	8.56	12.65	0.78	0.72	0.96	0.82
	PhysORD [42]	2.35	2.75	2.17	2.42	17.96	12.64	8.54	13.05	0.77	0.74	1.00	0.84
	AnyNav [12]	2.60	2.46	2.49	2.52	21.58	12.16	8.65	14.13	0.97	0.81	1.01	0.93
	Ours	2.30	0.06	0.05	0.81	18.25	9.21	8.22	11.90	0.77	0.87	0.78	0.81
Long Ditch	PO-Trav	2.74	3.17	3.77	3.22	16.87	13.66	6.92	12.49	0.91	0.83	1.07	0.94
	PhysORD [42]	2.39	3.27	3.91	3.19	22.66	13.13	7.12	14.30	0.81	0.88	1.09	0.93
	AnyNav [12]	2.51	3.10	4.01	3.21	19.76	12.80	8.49	13.69	0.83	0.79	1.14	0.92
	Ours	0.01	3.62	0.12	1.25	17.07	12.79	5.97	11.94	0.74	0.87	1.06	0.89
Bump	PO-Trav	2.52	2.04	2.94	2.50	17.91	16.29	7.75	13.98	0.87	0.92	1.07	0.95
	PhysORD [42]	2.98	2.17	3.35	2.83	20.15	14.02	7.88	14.02	0.85	0.79	1.05	0.90
	AnyNav [12]	2.49	2.45	2.93	2.62	19.35	14.79	7.29	13.81	0.80	0.84	1.03	0.89
	Ours	0.07	0.06	0.15	0.09	17.45	13.71	8.41	13.19	0.79	0.76	0.98	0.84

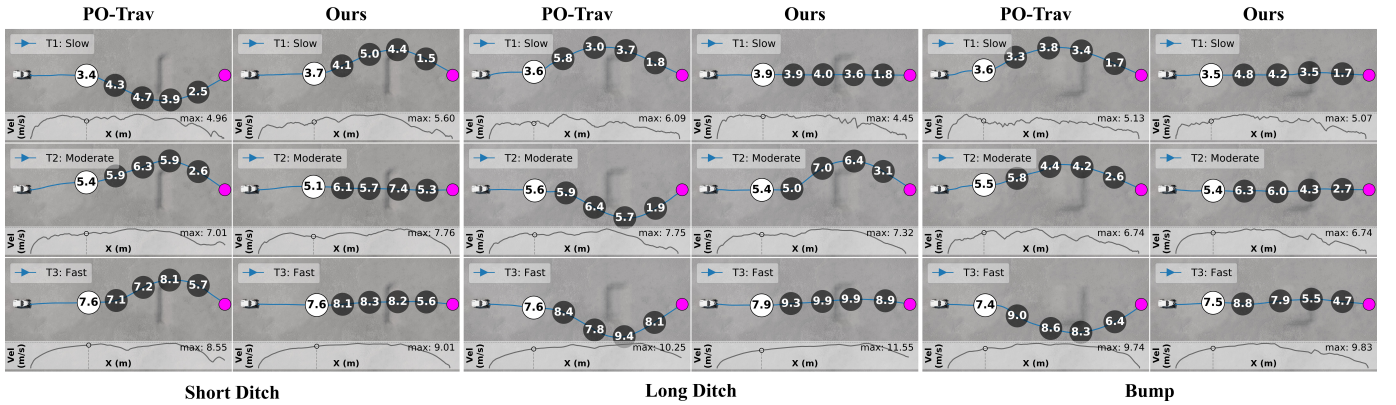


Fig. 5: Qualitative comparison between MAT and the PO-Trav in the obstacle traversal task. For each method, the vehicle trajectories with annotated velocities are shown for three obstacle types and three operating speeds. The velocity at which the obstacle first enters perception range is highlighted as a reference, and the full velocity profile is plotted for each test.

scenes, this metric effectively compares planning efficiency within the same scene, especially involving open areas. Traversal time measures the total time taken and reflects the ability to maintain safe and efficient speeds. Energy consumption, computed as integrated throttle effort, captures the control workload required to complete the trajectory and serves as a proxy for overall power efficiency. Together, these metrics capture the agility, safety, and efficiency of a navigation policy.

B. Simulation: Obstacle Traversal

To evaluate the ability of MAT to handle challenging obstacles, we consider a task in which the vehicle must find a feasible motion plan from a fixed start to a goal across obstacles. To emulate real-world conditions where a vehicle may encounter different obstacles at varying speeds, we test across two dimensions: (1) terrain geometry, including a short ditch, a long ditch and a bump; and (2) operating velocity, controlled by setting different MPPI time horizons $[T_1, T_2, T_3] = [5, 3, 2]$ seconds, corresponding to slow, moderate, and fast motion. Under this setup, the only difference between each comparison pair lies in whether the terrain cost is conditioned on velocity.

1) *Quantitative Comparison:* As summarized in Table I, MAT reduces the detour distance by an average of 75% across all obstacles compared to PO-Trav. MAT also achieves lower traversal time and energy consumption than AnyNav and PhysORD, as their dynamics models remain insufficient to model the coupled effects of terrain geometry and velocity. Across different velocity levels, the gains in the averaged metrics primarily occur in scenarios where agile yet safe motion plans are possible and enabled by the MAT, as reflected in the per-velocity results and illustrated qualitatively next.

2) *Qualitative Comparison:* Fig. 5 presents vehicle trajectories of MAT and PO-Trav, as AnyNav and PhysORD exhibit behaviors similar to PO-Trav. Overall, MAT produces more adaptive and efficient motion plans than PO-Trav. For the short ditch, PO-Trav always avoids it due to its consistently high terrain cost. In contrast, MAT avoids the ditch when the vehicle moves slowly and accelerates to jump across it when sufficient speed can be achieved. This flexibility arises because MAT first reasons about the velocities under current dynamics, then evaluates the terrain cost conditioned on those velocities. The Gaussian representation plays a significant role in enabling



Fig. 6: Simulation maps and qualitative results for the long-range navigation task. The top row shows three modified maps with introduced obstacles blocking the routes. The bottom row presents vehicle trajectories under MAT and the PO-Trav that exhibit distinct behavioral differences, with the corresponding obstacles highlighted in bold on the maps.

efficient terrain cost updates for new sampled velocities within the planner optimization loop. This principle generalizes to other geometries. For the long ditch, which requires a higher speed to clear, MAT adapts its strategy: crawling over at low speed, avoiding at medium speed, and accelerating to jump at high speed. For the bump, which is only traversable at low velocity, MAT chooses to decelerate before crossing. These behaviors emerge from different Gaussian profiles (A, μ, σ) predicted from perception. In contrast, PO-Trav exhibits uniform avoidance due to the underlying limitation of position-only terrain cost, resulting in suboptimal plans.

C. Simulation: Long-range Navigation

To evaluate MAT in realistic, large-scale scenarios, we conduct long-range navigation experiments in BeamNG, where unstructured obstacles are added to appear unexpectedly and block the nominal route. In all experiments, the vehicle follows global waypoints spaced about 12m apart, which do not account for obstacles or vehicle dynamics. Thus, the vehicle must rapidly adjust speed and trajectory to maintain safety.

1) *Navigation with Emerging Obstacles:* We first evaluate MAT on a 1084m forest map. To increase complexity, we add obstacles such as ditches, rocks, and bumps across uphill, downhill, and curved regions (see Fig. 6: Forest).

The vehicle must autonomously reach the final goal by continually planning motions toward each waypoint while handling encountered obstacles in real time. The resulting trajectories are shown in the numbered comparisons in Fig. 6. Compared with PO-Trav, MAT enables agile motion strategies, such as jumping a ditch in comparison (1). Moreover, MAT adapts its behaviors to the vehicle’s motion dynamics. For instance, comparison (2) illustrates two contrasting scenarios: MAT avoids a ditch when approaching uphill with insufficient speed, but accelerates to jump across a downhill ditch when the momentum gained during descent makes the maneuver feasible. This experiment showcases MAT’s ability to account for speed-dependent traversability and respond effectively to emerging obstacles during long-range forest navigation.

2) *Generalization to Unseen Maps:* To assess generalization to unseen environments, we create two additional maps modified from the Utah Canyon (638m) and Desert (634m) environments. These maps introduce new terrain geometries, such as boulders and bedrocks, and distinct vehicle-terrain interactions, including sand surfaces. Across both unseen environments, MAT preserves its agile and adaptive behavior. It reproduces the ditch-handling strategies observed in the forest experiment, as illustrated in comparison (3). MAT also responds appropriately to new obstacle types, as shown in

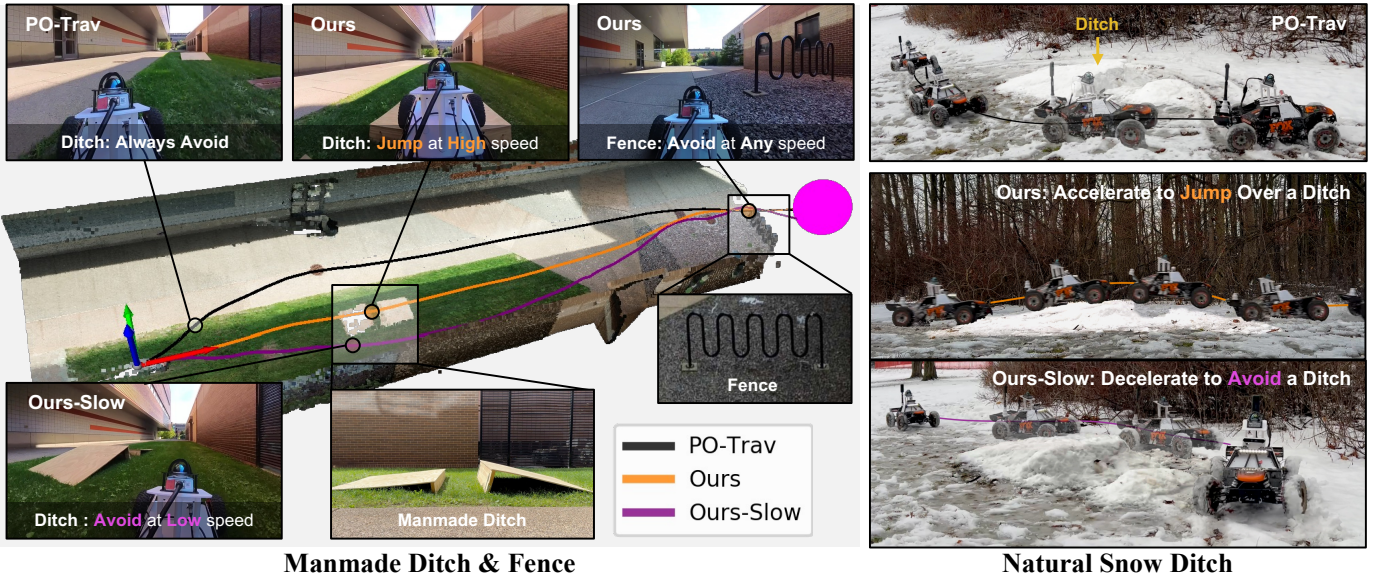


Fig. 7: Real-world ditch traversal strategies in manmade (left) and natural (right) environments. While PO-Trav is consistently conservative, MAT demonstrates velocity-dependent reasoning by jumping (Ours) or avoiding (Ours-Slow). Simultaneously, MAT identifies the fence as universally non-traversable. Overlaid trajectories illustrate these distinct motion strategies.

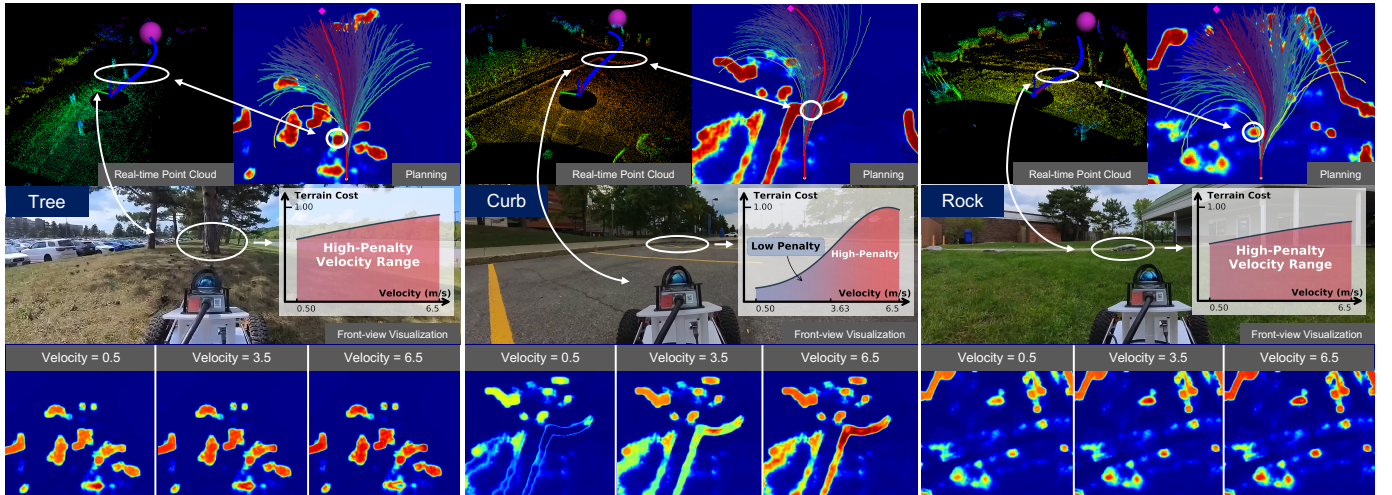


Fig. 8: Natural obstacles encountered in real-world navigation: tree, curb, and rock. Top: LiDAR points, sampled MPPI trajectories (where path costs are colored from blue/low to yellow/high and the goal is in magenta), and the predicted peak penalty map ($A(\mathbf{x})$, defined in Eq. 1). Middle: The predicted velocity-dependent cost function for a selected obstacle pixel and the third-person front view of the obstacle. Bottom: The resulting traversability maps when all pixels are evaluated at three constant velocities (0.5 m/s, 3.5 m/s, and 6.5 m/s), with map costs colored from blue (low) to red (high).

comparison (4): it safely avoids untraversable boulders and rocks and decelerates to crawl over flatter bedrock surfaces.

Together, comparisons (1) through (4) highlight MAT’s ability to rapidly and robustly adapt its motion strategy to both vehicle dynamics and unstructured terrain geometries.

D. Real-world: Ditch Traversal

To evaluate the real-world performance of MAT, we deploy our system on the *Racer* in two ditch traversal experiments: a manmade environment and a natural field. Both scenarios

involve obstacles with significant geometric challenges that are typically considered untraversable by prior methods, forcing traditional planners to take long detours. We compare MAT with PO-Trav, as AnyNav and PhysORD fail to produce stable results when deployed on *Racer* for these scenarios.

a) *Manmade Ditch & Fence*: In the manmade setting, the vehicle navigates toward a goal blocked by a ditch and a tall metal fence. We visualize the resulting trajectories in Fig. 7, where the environment is reconstructed using COLMAP [25, 26]. The PO-Trav consistently avoids both obstacles due to the

high terrain cost, leading to substantial detours. In contrast, MAT identifies that the ditch is traversable at high speed and accelerates to jump across it, while correctly recognizing the fence as untraversable at all velocities. When using the same MAT model with planner parameters adjusted to simulate a slow-moving vehicle (“Ours-Slow”), MAT correctly identifies that it cannot reach the velocity required for a safe jump and shifts its strategy to avoid the ditch. This highlights its ability to perform robust reasoning under current system dynamics.

TABLE II: Comparison in Real-world Ditch Traversal.

Scenario	Method	Detour Distance [m] ↓	Traversal Time [s] ↓	Energy Consumption ↓
Manmade Ditch & Fence	PO-Trav	1.54	15.40	6.13
	Ours	0.49	13.51	5.19
Natural Snow Ditch	PO-Trav	1.58	5.01	1.95
	Ours	0.09	2.01	0.81

b) Natural Snow Ditch: In the natural scenario, the vehicle must traverse an uneven, snow-filled ditch. Trajectories are visualized using sequential vehicle overlays to illustrate the traversal history. As in the manmade case, the PO-Trav avoids the ditch entirely, whereas MAT exploits higher speeds to traverse it directly and avoids it when operating under low-speed conditions that make a safe jump infeasible.

Across both experiments, MAT quantitatively outperforms the PO-Trav across all metrics. For a 40 m navigation task in the manmade setting, MAT reduces detour distance by 68.2% compared to the PO-Trav. In the natural ditch experiment, MAT completes the traversal in 40.1% of the time required by the PO-Trav. MAT also achieves lower energy consumption due to more efficient motion plans in both scenarios.

E. Real-world: Outdoor Navigation with Obstacles

We evaluate our system during autonomous navigation among diverse natural obstacles in the outdoor environment. Fig. 8 shows obstacles encountered, including trees, curbs, and rocks, and the perception, MAT map prediction, and planning results in our navigation system. Each obstacle type exhibits distinct traversability-velocity characteristics: trees and rocks maintain high terrain costs across all speeds, while curbs have low costs at low speed but high costs at high speed. The MAT map captures these differences through varying Gaussian parameters (A, μ, σ), enabling the MPPI planner to generate adaptive maneuvers: avoiding trees and rocks while decelerating to crawl over curbs. These results demonstrate MAT’s effectiveness in handling varied obstacles outdoors.

V. CONCLUSIONS AND FUTURE WORK

This paper introduced Motion-Aware Traversability (MAT), a new representation that conditions terrain cost on both terrain perception and vehicle motion for off-road navigation. By modeling traversability as a velocity-dependent function with parameters learned from perception, MAT captures how motion feasibility varies across terrains and enables efficient cost evaluation during planning. Experiments in BeamNG simulation and real-world settings show that MAT improves

adaptability and efficiency, enabling vehicles to plan agile yet safe maneuvers such as accelerating to jump across ditches, slowing on bumps and curbs, and consistently avoiding untraversable obstacles. While the Gaussian function enables a step forward from previous traversability maps, it remains a simplified assumption. Furthermore, a primary system failure arises from complete LiDAR occlusion, where the system becomes overly conservative, but a higher sensor placement would resolve this issue. Future work will extend MAT beyond the Gaussian model and incorporate additional motion factors to enhance robustness and expressiveness.

ACKNOWLEDGEMENTS

This work was partially funded by ONR award N00014-24-1-2003 and DARPA award HR00112490426. 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 ONR or DARPA. The authors also thank Yi Du and Zitong Zhan (University at Buffalo) for technical support and insightful discussions.

REFERENCES

- [1] BeamNG GmbH. BeamNG.tech. URL <https://www.beamng.tech/>.
- [2] Paulo VK Borges, Thierry Peynot, Sisi Liang, Bilal Arain, Matthew Wildie, Melih G 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*, 2:1567–1627, 2022.
- [3] 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. *IEEE Robotics and Automation Letters*, 2025.
- [4] Chao Cao, Hongbiao Zhu, Fan Yang, Yukun Xia, Howie Choset, Jean Oh, and Ji Zhang. Autonomous exploration development environment and the planning algorithms. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8921–8928. IEEE, 2022.
- [5] Mateo Guaman Castro, Samuel Triest, Wenshan Wang, Jason M Gregory, Felix Sanchez, John G Rogers, and Sebastian Scherer. How does it feel? self-supervised costmap learning for off-road vehicle traversability. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 931–938, 2023.
- [6] Eric Chen, Cherie Ho, Mukhtar Maulimov, Chen Wang, and Sebastian Scherer. Learning-on-the-drive: Self-supervised adaptive long-range perception for high-speed offroad driving. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024. URL <https://arxiv.org/abs/2306.15226>.
- [7] Aniket Datar, Chenhui Pan, Anuj Pokhrel, Matthew Choulas, Mohammad Nazeri, Tong Xu, and Xuesu Xiao. Traverse the non-traversable: Estimating traversability for wheeled mobility on vertically challenging terrain. In *RSS 2025 Workshop on Resilient Off-road Autonomous Robotics*.
- [8] Aniket Datar, Chenhui Pan, Mohammad Nazeri, Anuj Pokhrel, and Xuesu Xiao. Terrain-attentive learning for efficient 6-dof kinodynamic modeling on vertically challenging terrain. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5438–5443. IEEE, 2024.
- [9] Aniket Datar, Chenhui Pan, and Xuesu Xiao. Learning to model and plan for wheeled mobility on vertically challenging terrain. *IEEE Robotics and Automation Letters*, 2024.

- [10] Anushri Dixit, Ali-akbar Agha-mohammadi, David D Fan, Joel W Burdick, Yuki Kubo, and Kyohei Otsu. Step: Stochastic traversability evaluation and planning for risk-aware off-road navigation. *Robotics: Science and Systems XVII*, 2021.
- [11] Jonas Frey, Matias Mattamala, Nived Chebrolu, Cesar Cadena, Maurice Fallon, and Marco Hutter. Fast Traversability Estimation for Wild Visual Navigation. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.054.
- [12] Taimeng Fu, Zitong Zhan, Zhipeng Zhao, Shaoshu Su, Xiao Lin, Ehsan Tarkesh Esfahani, Karthik Dantu, Souma Chowdhury, and Chen Wang. Anynav: Visual neuro-symbolic friction learning for off-road navigation. *arXiv preprint arXiv:2501.12654*, 2025.
- [13] Mateus V Gasparino, Arun N Sivakumar, and Girish Chowdhary. Wayfaster: a self-supervised traversability prediction for increased navigation awareness. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8486–8492. IEEE, 2024.
- [14] Tyler Han, Alex Liu, Anqi Li, Alex Spitzer, Guanya Shi, and Byron Boots. Model predictive control for aggressive driving over uneven terrain. *arXiv preprint arXiv:2311.12284*, 2023.
- [15] Sanghun Jung, JoonHo Lee, Xiangyun Meng, Byron Boots, and Alexander Lambert. V-strong: Visual self-supervised traversability learning for off-road navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1766–1773. IEEE, 2024.
- [16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.
- [17] 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.
- [18] 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.
- [19] Hojin Lee, Taekyung Kim, Jungwi Mun, and Wonsuk Lee. Learning terrain-aware kinodynamic model for autonomous off-road rally driving with model predictive path integral control. *IEEE Robotics and Automation Letters*, 8(11):7663–7670, 2023.
- [20] Xiangyun Meng, Nathan Hatch, Alexander Lambert, Anqi Li, Nolan Wagener, Matthew Schmittle, Joonho Lee, Wentao Yuan, Zoey Qiuyu Chen, Samuel Deng, et al. Terrainnet: Visual modeling of complex terrain for high-speed, off-road navigation. In *Robotics: Science and Systems*, 2023.
- [21] Mohammad Nazeri, Anuj Pokhrel, Alexandyr Card, Aniket Datar, Garrett Warnell, and Xuesu Xiao. Vertiformer: A data-efficient multi-task transformer for off-road robot mobility. *arXiv preprint arXiv:2502.00543*, 2025.
- [22] Anuj Pokhrel, Aniket Datar, and Xuesu Xiao. Dom, cars don’t fly!—or do they? in-air vehicle maneuver for high-speed off-road navigation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 21325–21332. IEEE, 2025.
- [23] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, 2009.
- [24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [25] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [26] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.
- [27] Junwon Seo, Taekyung Kim, Kiho Kwak, Jihong Min, and Inwook Shim. Scate: A scalable framework for self-supervised traversability estimation in unstructured environments. *IEEE Robotics and Automation Letters*, 8(2):888–895, 2023.
- [28] Junwon Seo, Sungdae Sim, and Inwook Shim. Learning off-road terrain traversability with self-supervisions only. *IEEE Robotics and Automation Letters*, 8(8):4617–4624, 2023.
- [29] Junwon Seo, Taekyung Kim, Seongyong Ahn, and Kiho Kwak. Metaverse: Meta-learning traversability cost map for off-road navigation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13190–13197, 2024.
- [30] 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. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16999–17006. IEEE, 2025.
- [31] 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*.
- [32] 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. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 924–930. IEEE, 2023.
- [33] Bogdan Vlahov, Jason Gibson, Manan Gandhi, and Evangelos A Theodorou. Mppi-generic: A cuda library for stochastic trajectory optimization. *arXiv preprint arXiv:2409.07563*, 2024.
- [34] Chen Wang, Dasong Gao, Kuan Xu, Junyi Geng, Yaoyu Hu, Yuheng Qiu, Bowen Li, Fan Yang, Brady Moon, Abhinav Pandey, Aryan, Jiahe Xu, Tianhao Wu, Haonan He, Daning Huang, Zhongqiang Ren, Shibo Zhao, Taimeng Fu, Pranay Reddy, Xiao Lin, Wenshan Wang, Jingnan Shi, Rajat Talak, Kun Cao, Yi Du, Han Wang, Huai Yu, Shanzhao Wang, Siyu Chen, Ananth Kashyap, Rohan Bandaru, Karthik Dantu, Jiajun Wu, Lihua Xie, Luca Carlone, Marco Hutter, and Sebastian Scherer. PyPose: A library for robot learning with physics-based optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22024–22034, 2023. URL <https://arxiv.org/abs/2209.15428>.
- [35] Chen Wang, Kaiyi Ji, Junyi Geng, Zhongqiang Ren, Taimeng Fu, Fan Yang, Yifan Guo, Haonan He, Xiangyu Chen, Zitong Zhan, Qiwei Du, Shaoshu Su, Bowen Li, Yuheng Qiu, Yi Du, Qihang Li, Yifan Yang, Xiao Lin, and Zhipeng Zhao. Imperative learning: A self-supervised neuro-symbolic learning framework for robot autonomy. *International Journal of Robotics Research (IJRR)*, 2025. URL <https://arxiv.org/abs/2406.16087>.
- [36] 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.
- [37] Grady Williams, Paul Drews, Brian Goldfain, James M Rehg, and Evangelos A Theodorou. Aggressive driving with model predictive path integral control. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1433–1440. IEEE, 2016.
- [38] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lid2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 38(4):2053–2073, 2022.

- [39] Yuxiang Yang, Xiangyun Meng, Wenhao Yu, Tingnan Zhang, Jie Tan, and Byron Boots. Learning semantics-aware locomotion skills from human demonstration. In *Conference on robot learning*, pages 2205–2214. PMLR, 2023.
- [40] Ziwon Yoon, Lawrence Y Zhu, Jingxi Lu, Lu Gan, and Ye Zhao. State-nav: Stability-aware traversability estimation for bipedal navigation on rough terrain. *IEEE Robotics and Automation Letters*, 11(2):2338–2345, 2025.
- [41] Zitong Zhan, Xiangfu Li, Qihang Li, Haonan He, Abhinav Pandey, Haitao Xiao, Yangmengfei Xu, Xiangyu Chen, Kuan Xu, Kun Cao, Zhipeng Zhao, Zihan Wang, Huan Xu, Zihang Fang, Yutian Chen, Wentao Wang, Xu Fang, Yi Du, Tianhao Wu, Xiao Lin, Yuheng Qiu, Fan Yang, Jingnan Shi, Shaoshu Su, Yiren Lu, Taimeng Fu, Karthik Dantu, Jiajun Wu, Lihua Xie, Marco Hutter, Luca Carlone, Sebastian Scherer, Daning Huang, Yaoyu Hu, Junyi Geng, and Chen Wang. PyPose v0.6: The imperative programming interface for robotics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Workshop*, 2023. URL <https://arxiv.org/abs/2309.13035>.
- [42] Zhipeng Zhao, Bowen Li, Yi Du, Taimeng Fu, and Chen Wang. Physord: a neuro-symbolic approach for physics-infused motion prediction in off-road driving. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11670–11677. IEEE, 2024.