

# STATE-NAV: Stability-Aware Traversability Estimation for Bipedal Navigation on Rough Terrain

Ziwon Yoon , Lawrence Y. Zhu , Jingxi Lu, Lu Gan , and Ye Zhao 

**Abstract**—Bipedal robots have advantages in maneuvering human-centered environments, but face greater failure risk compared to other stable mobile platforms, such as wheeled or quadrupedal robots. While learning-based traversability has been widely studied for these platforms, bipedal traversability has instead relied on manually designed rules with limited consideration of locomotion stability on rough terrain. In this work, we present the first learning-based traversability estimation and risk-sensitive navigation framework for bipedal robots operating in diverse, uneven environments. TravFormer, a transformer-based neural network, is trained to predict bipedal instability with uncertainty, enabling risk-aware and adaptive planning. Based on the network, we define traversability as stability-aware command velocity—the fastest command velocity that keeps instability below a user-defined limit. This velocity-based traversability is integrated into a hierarchical planner that combines traversability-informed Rapid Random Tree Star (TravRRT\*) for time-efficient path planning and Model Predictive Control (MPC) for safe execution. We validate our method in MuJoCo simulator and the real world, demonstrating improved stability, time efficiency, and robustness across diverse terrains compared with existing methods.

**Index Terms**—Humanoids, legged robots, traversability, navigation, planning, model predictive control, stability.

## I. INTRODUCTION

With the rapid advancement of legged robotics, humanoid robots have garnered significant attention for their ability to navigate complex environments with rough terrain that are often inaccessible to wheeled mobile systems [1], [2]. Despite their advantages, humanoid robots face significant challenges in maintaining stability and balance, particularly on uneven or dynamically changing terrain [3]. Unlike wheeled or quadrupedal robots, which benefit from their more statically stable ground

contacts, humanoids have a higher risk of failure due to their limited support area and elevated center of mass, especially during rapid locomotion or abrupt terrain transitions. While various control methods have been proposed to address locomotion stability [4], [5], traversability and navigation frameworks that account for the instability of bipedal locomotion on rough terrains remain largely underexplored.

Existing works have explored bipedal traversability using manually designed rules based on geometry [6], [7], [8], providing a rough representation of terrain difficulty. However, these manually designed rules often fail to accurately reflect locomotion stability and performance. Meanwhile, for wheeled and quadrupedal robots, extensive research on traversability estimation has leveraged self-supervised learning with various locomotion features such as traction or IMU variances [9], [10], [11], [12], [13], [14], [15], [16]. However, the relevance of these features to humanoid locomotion stability remains uncertain due to the fundamentally different locomotion strategies employed by bipedal robots. Furthermore, the greater risk of failure in humanoid locomotion requires them to operate under stricter constraints on motion aggressiveness, such as command velocity. This aspect is often overlooked in traditional navigation planning but is crucial for ensuring safe and efficient humanoid navigation.

To address these problems, we take a data-driven approach to evaluate various locomotion features and identify the one that best correlates with bipedal instability for predicting fallover risk. A Transformer-based neural network, TravFormer, is then trained to predict the identified metric from terrain features and commanded velocity using simulated data. Instead of directly using the predicted instability as a traversability measure, we define traversability as the *stability-aware command velocity*, i.e., the fastest command velocity that maintains predicted instability below a user-defined threshold, which we use to construct a traversability map over the environment, as illustrated in Fig. 1.

Estimating traversability as a stability-aware command velocity enables stability-prioritized navigation planning. Many traditional planning approaches take traversability as a cost term with tunable weights and jointly optimize it with other costs such as the path length [6], [11], [14], [16], [17], [18]. However, varying path length scales across environments often necessitate re-tuning of weights. On the other hand, using stability-aware command velocity mitigates the effort of manual weight tuning, as terrain difficulty is intrinsically encoded in the traversal time via a robot-specific instability threshold instead of environment-specific weights. We leverage this representation in a hierarchical Traversability-Informed Rapidly-exploring Random Tree Star (TravRRT\*)–Model Predictive Control (MPC) navigation planning algorithm as shown in Fig. 1. The global TravRRT\* planner simplifies the planning objective to navigation time

Received 15 June 2025; accepted 19 November 2025. Date of publication 26 December 2025; date of current version 19 January 2026. This article was recommended for publication by Associate Editor L. Rossini and Editor A. Kheddar upon evaluation of the reviewers' comments. This work was supported in part by the Office of Naval Research under Grant N000142312223, in part by the National Science Foundation (NSF) under Grant CMMI-2144309 and Grant FRR-2328254, and in part by the United States Department of Agriculture (USDA) under Grant 2023-67021-41397. (Corresponding author: Ye Zhao.)

Ziwon Yoon, Lawrence Y. Zhu, Lu Gan, and Ye Zhao are with the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: zyoone6@gatech.edu; lawrencezhu@gatech.edu; lgan@gatech.edu; yezhao@gatech.edu).

Jingxi Lu is with the Georgia Institute of Technology, Atlanta, GA 30332 USA, and also with the Electrical and Computer Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: jingxil@usc.edu).

Project page with code: <https://state-nav.github.io/statenav/>.

This article has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2025.3648502>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3648502

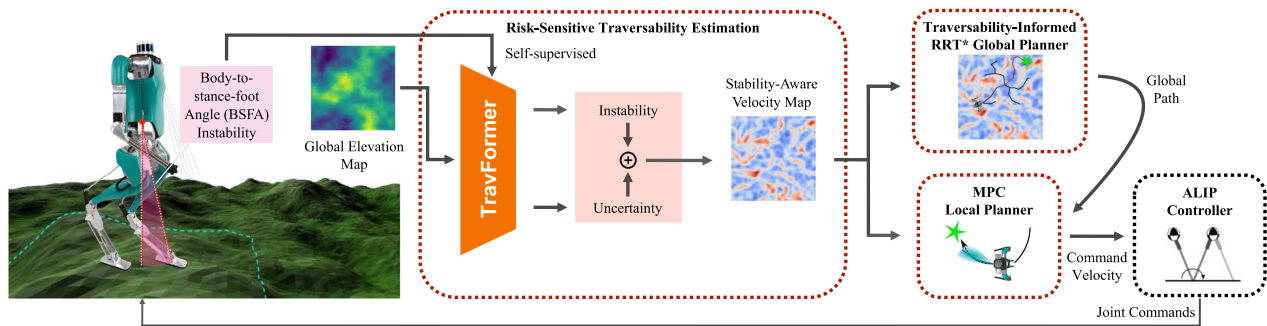


Fig. 1. Overall diagram of the proposed traversability estimation and the navigation framework. A transformer-based bipedal instability estimator, *TravFormer*, is trained using body-to-stance-foot angle (BSFA) instability as a self-supervised signal to generate stability-aware command velocity map from the geometric representation of the environment. This map is further used by a hierarchically integrated *TravRRT\** global planner and MPC local planner for safe navigation.

optimization, while leveraging terrain traversability to guide informed sampling toward safer, more navigable regions. Additionally, the learned command is incorporated as a constraint in the Linear Inverted Pendulum Model (LIPM)-based local MPC planner, to generate an optimal motion plan that follows the global path while satisfying stability constraints.

In summary, we propose a learning-based bipedal traversability estimation framework that formally incorporates locomotion stability, and design a hierarchical planning framework tailored specifically for humanoid navigation. The key contributions of this letter are summarized as follows:

- We conduct a comparative analysis of locomotion features to determine the best feature reflecting bipedal instability—the body-to-stance-foot angle (BSFA)—and use it as the supervisory signal guiding the neural network.
- *TravFormer* is introduced as the first learning-based traversability estimator for humanoid robots on diverse rough terrain, predicting instability with uncertainty from self-supervised BSFA labels.
- Stable bipedal navigation across diverse environments is achieved by representing traversability as a stability-aware command velocity, providing robust, consistent performance without extensive gain tuning when integrated into the hierarchical *TravRRT\**-MPC framework.
- We validate the full framework on the Digit humanoid in MuJoCo simulator [19] and real-world tests, demonstrating safety and efficiency on diverse rough terrain.

## II. RELATED WORK

### A. Self-Supervised Traversability Learning for Mobile Robots

Traversability characterizes a robot’s ability to navigate its environment by quantifying terrain difficulty [20]. In existing methods, traversability has been estimated based on manually defined rules based on terrain geometry [18] or supervised learning with human-labeled semantics [21]. However, both approaches often fail to capture the complex interaction between robots and environments. More recent efforts have developed learning-based traversability estimators in a self-supervised manner, using supervisory signals derived from various proprioceptive features during locomotion, such as ground reaction wrench analysis [9], foothold scores [10], traction [11], [12], [13], statistical features of sensor measurements such as IMU data [14], and energy consumption metric [15], [16]. While these self-supervising features capture specific aspects of locomotion

difficulty of wheeled and quadrupedal robots, they lack validation regarding their ability to represent the fallover risk in case of bipedal locomotion.

### B. Traversability Estimation and Navigation Planning for Bipedal Locomotion

Traversability for bipedal locomotion on rough terrain remains largely underexplored. In [6], A\* planning samples candidate footsteps around nodes and defines traversability as the percentage of cells with sufficiently low inclination and height proximity within the sampled region. Other works have explored learning-based traversability estimation for traversing tilted flat surfaces based on footstep availability [22], [23], or adjusting body height to navigate low-ceiling environments [24]. These methods target specific settings and do not address large-scale navigation across general rough terrain.

### C. Navigation Planning With Traversability Estimation

Autonomous navigation systems use traversability at multiple levels of the planning stack to navigate safely. A common approach at the global planning level is to define traversability as a *score* or *cost* and combine it with other costs such as path length, trading off between locomotion stability and navigation time [6], [11], [14], [16], [17], [18]. While incorporating traversability, the works in [12], [13], [17] use Conditional Value at Risk (CVaR) to incorporate environmental uncertainty into the planning for risk-sensitive navigation. At the local planning level, legged robots have extensively employed MPC to handle complex kinodynamic constraints and generate dynamically feasible motion plans. Depth-integrated MPC incorporates terrain geometry to guide footstep placement on uneven ground [25], [26]. Other works [8], [17] incorporate traversability costs into MPC for further refining the path plan from a global planner. However, these approaches typically fix command velocities and overlook high-level selection of velocity commands.

## III. INSTABILITY LEARNING FOR SAFE BIPEDAL NAVIGATION PLANNING

### A. Instability and Fallover Prediction

To identify an effective self-supervised signal to train our learning-based traversability estimator specifically for bipedal navigation, we begin by analyzing various locomotion features for their correlations to fallover risk in bipedal locomotion.

TABLE I  
COMPARISON OF MCFADDEN'S  $R^2$  AND AUC-ROC SCORES FOR VARIOUS LOCOMOTION FEATURES, EVALUATING THEIR EFFECTIVENESS IN PREDICTING FALLOVER RISK. THE HIGHEST VALUES IN EACH COLUMN ARE SHOWN IN BOLD

	McFadden's $R^2$	AUC-ROC
Energy (Control Effort) [15]	0.184	0.784
IMU (PSD) [14]	0.217	0.837
Traction [11], [12]	0.097	0.770
Tangential/Normal Force Ratio	0.016	0.806
Center of Pressure (x-dir)	0.025	0.859
Center of Pressure (y-dir)	0.059	0.879
Body-to-stance-foot angle (Ours)	<b>0.356</b>	<b>0.906</b>

We collect walking data from Digit in MuJoCo on randomly generated rough terrains with varying geometries, such as slopes, bumps, and surface roughness. Digit is controlled using a terrain-aware ALIP controller [8], commanded with varying linear and angular velocities. The collected data is then parsed into gait cycles, and locomotion features are computed as the Root-Mean-Square (RMS) of raw signals per cycle. Each gait cycle is then labeled with a fallover event: 1 if a fall occurs within two subsequent steps, and 0 otherwise. All candidate features are normalized by their RMS during in-place stepping on flat ground prior to correlation analysis and training.

With this data, we evaluate the correlation between each feature and fallover events via logistic regression. Table I summarizes the results using McFadden's  $R^2$  [27] and AUC-ROC score [28], where  $R^2$  around 0.4 and a score close to 1 generally indicates a strong relationship. While features commonly used in grounded mobile robots, such as IMU signals and traction, show acceptable correlations, the body-to-stance-foot angle (BSFA) described in Fig. 1 leads to better prediction accuracy. This result is reasonable because the bipedal robot counteracts its body perturbation by placing its foot further from the body to increase control authority [5]. We refer to this per-cycle RMS as *BSFA instability*  $\delta$ . Moreover, based on our dataset, we empirically found that BSFA instability is only marginally influenced by slight variations in foot placement, indicating that the metric is robust to small contact-location errors. Based on this finding, we adopt the BSFA instability as the supervisory signal for training our traversability estimation network to predict bipedal instability.

### B. Learning Instability

We aim to predict the BSFA instability from the terrain and the robot state. In this work, we represent the geometrical features of the environment using a 2.5D elevation map [29]. For the robot state, many components such as instant joint values are highly time-varying and hence difficult to consider as decision variables in planning. However, other states, such as velocity action commands, are relatively less time-varying and play a critical role when the learned traversability is later exploited in planning. Hence, we define our instability estimation model as follows:

$$\delta = p_{\theta_{\delta}}(m_{\text{ego}}, a), \quad (1)$$

where  $p_{\theta_{\delta}}(\cdot)$  is the target function that maps  $m_{\text{ego}}$  and  $a$  to  $\delta$ . More specifically,  $m_{\text{ego}}$  is a  $0.64\text{m} \times 0.64\text{m}$  robot-centric (yaw-aligned) elevation map patch.  $a$  is a command vector

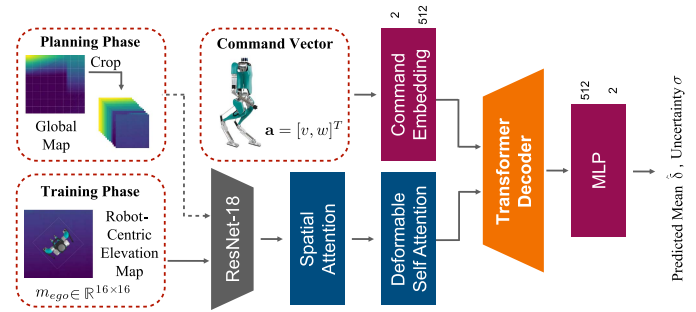


Fig. 2. TravFormer architecture combines convolutional feature extraction with attention mechanisms to predict BSFA instability. The model processes an elevation map patch  $m_{\text{ego}}$  representing local terrain and a command vector  $a = [v, w]^T$ . During training, the model uses robot-centric map patches centered on the robot. At planning time, the global terrain map is divided into multiple local patches, which are processed in batch.

$(v, w)^T$  with linear velocity  $v$  and angular velocity  $w$ . We model  $\delta$  as a Gaussian distribution with mean  $\hat{\delta}$  and standard deviation  $\sigma$ , where  $\sigma$  represents the aleatoric uncertainty arising from perception noise and unmodeled effects of instantaneous joint and robot states, and is subsequently used for risk-sensitive planning. Note that external disturbances introduce unmodeled perturbations to the controller, analogous to sensor noise and mapping errors, and are therefore implicitly captured within the aleatoric uncertainty.

To model this function, we propose *TravFormer*, a neural network architecture designed to predict BSFA instability from the elevation map and command, as shown in Fig. 2. TravFormer leverages a ResNet-18 backbone [30] for terrain feature extraction and incorporates attention mechanisms [31] to focus on stability-critical aspects of the terrain while enabling context-aware integration of the command inputs. The Transformer Decoder treats command embeddings as queries and terrain features as keys/values to generate the final traversability predictions. We use a two-phase schedule for stable joint optimization of  $\hat{\delta}$  and  $\sigma$ , inspired from [32]

**Phase 1 - Mean Squared Error (MSE) Training:** In the initial phase, we train the network to minimize the MSE between the network prediction of BSFA instability and the labels collected in Section III-A. We train for 10 epochs during this phase, which we find sufficient to establish a strong foundation to ensure convergence for the subsequent uncertainty learning.

**Phase 2 - Gaussian Negative Log-Likelihood (NLL) Training [33]:** After initializing with weights from the MSE phase, we extend the network with an additional output head to predict both the mean  $\hat{\delta}$  and standard deviation  $\sigma$ . The network is trained to match the predicted BSFA instability distribution with the label distribution under a Gaussian assumption:

$$\mathcal{L}_{\text{GaussianNLL}} = \frac{1}{N} \sum_{i=1}^N \left[ \frac{(\hat{\delta}_i - \delta_i)^2}{2 \exp(2 \log \sigma_i)} + \log \sigma_i \right], \quad (2)$$

where  $N$  is the total number of samples in a batch and  $\delta$  denotes the target label collected in Section III-A.

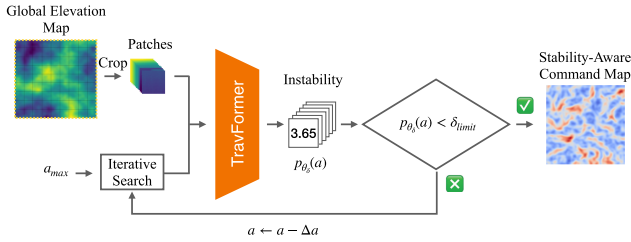


Fig. 3. Stability-aware command-velocity selection via iterative search. Starting from the maximum candidate velocity  $a_{\max}$ , the Iterative Search block feeds  $a_{\max}$  into the TravFormer model and then successively replaces it with the next lower candidate sweeping downward until the risk-sensitive stability criterion is met.

TABLE II

RMSE AND  $|\Delta\text{PICP}|$  RESULTS ON 3 ABLATED VARIANTS AND THE FULL VERSION OF TRAVFORMER.  $|\Delta\text{PICP}|$  REPRESENTS THE ABSOLUTE DIFFERENCE BETWEEN THE PROPORTION OF GROUND TRUTH INSTABILITY VALUES FALL WITHIN ONE STANDARD DEVIATION OF THE PREDICTED RANGE (CONSTRUCTED USING PREDICTED INSTABILITY VALUE  $\hat{\delta}$  AND STANDARD DEVIATION  $\sigma$ , ASSUMING GAUSSIAN DISTRIBUTION) AND THE IDEAL COVERAGE OF 68.27%. THE BEST VALUES ARE MARKED IN BOLD

Variant	RMSE	$ \Delta\text{PICP} $
TravFormer (Full)	0.7065	<b>0.0007</b>
TravFormer w/o Spatial Attention	0.7025	0.0564
TravFormer w/o Deformable Self-Attention	<b>0.6962</b>	0.0098
TravFormer w/ MLP Decoder	0.7958	0.1571

## IV. BIPEDAL NAVIGATION PLANNING

### A. Traversability as Stability-Aware Command Velocity

While prior work often treats traversability as a tunable cost term, such cases are less suitable for bipedal robots, where stability is critical. Instead, we define traversability as the fastest command velocity that satisfies a stability constraint, while accounting for uncertainty in a risk-sensitive manner. Given a user-defined instability limit  $\delta_{\text{limit}}$  and the risk level  $\alpha$ , we compute the risk-sensitive stability-aware command  $a^* = (v^*, w^*)$ , which characterizes the traversability of the given local map patch  $m_{\text{ego}}$ :

$$v_{m_{\text{ego}}}^* = \max \{v \mid \text{VaR}(p_{\theta_\delta}(m_{\text{ego}}, [v, 0]), \alpha) < \delta_{\text{limit}}\}, \quad (3)$$

$$\omega_{m_{\text{ego}}}^* = \max \{\omega \mid \text{VaR}(p_{\theta_\delta}(m_{\text{ego}}, [0, \omega]), \alpha) < \delta_{\text{limit}}\}, \quad (4)$$

where  $\text{VaR}(\cdot)$  denotes the Value at Risk [17], calculated from the predicted mean and variance under a Gaussian assumption. This fastest velocity is found approximately by sweeping the command velocity from its maximum candidate  $a_{\max} = (0.5 \text{ m/s}, 0.75 \text{ rad/s})$  downward to zero as in Fig. 3, with a resolution  $\Delta a = (0.05 \text{ m/s}, 0.075 \text{ rad/s})$ . For terrains where the predicted instability exceeds the threshold even at zero command velocity, we assign minimal values of 0.001m/s and 0.001 rds as the stability-aware command velocities, respectively, to penalize traversal through those regions. Note that  $\delta_{\text{limit}}$  is a robot-specific hyperparameter that does not need to be adjusted according to environmental conditions, but instead serves to configure the robot's stability during locomotion. In this letter, we use  $\delta_{\text{limit}} = 3$  and  $\alpha = 0.97$ . Representing traversability as the fastest command velocity under an instability constraint allows the planner to generate navigation-time-optimal paths

without weight tuning, while keeping instability within acceptable bounds. This velocity can also be used to constrain the local MPC planner, ensuring stable execution of the high-level navigation plan.

*Remark 1 (Use of risk-sensitive traversability):* We observe that as command velocity increases, the predicted instability grows in a superlinear manner with increasing uncertainty, particularly on steep terrain. By accounting for this behavior, VaR is incorporated to mitigate overestimation of safe velocities and promote stable navigation under uncertainty.

### B. Traversability-Informed RRT\* Global Planner

We propose traversability-informed RRT\* (TravRRT\*) built upon the works of [34], [35], [36] to generate a global path that minimizes navigation time while aiming to keep instability below a user-defined limit and risk level. During sample generation, the traversability-informed RRT\* samples nodes with probability proportional to local traversability, encouraging the search toward more navigable paths. For the cost calculation, the cost  $c_{ij}$  of an edge connecting a parent node  $n_i = (x_i, y_i)$  to a child node  $n_j = (x_j, y_j)$  is defined as the expected traversal time along the edge, subject to stability constraints:

$$c_{ij} = \sum_{k \in M} \left( \frac{\Delta l_k}{v_k^*} + \frac{\Delta \theta_k}{\omega_k^*} \right), \quad (5)$$

where  $M$  is the set of uniformly sampled intermediate points along the edge. The  $\Delta l_k$  and  $\Delta \theta_k$  are the linear displacement and heading change at point  $k$ , while  $v_k^*$  and  $\omega_k^*$  are the stability-aware linear and angular velocity commands for that point computed by (3), (4). By incorporating both translational and rotational traversal time at each intermediate point, this cost formulation allows the planner to select paths that are not only stable but also time-efficient, accounting for both translating and turning motions.

### C. MPC Local Planner

To ensure stable execution of the global path generated by the RRT\* planner, we formulate an MPC planner that tracks waypoints while accounting for the reduced-order bipedal walking dynamics. We begin with the LIPM to derive discrete dynamics in the local sagittal direction, assuming the fixed step duration  $T$  [5]. The CoM position change  $\Delta x_q^{\text{loc}}$  between the  $q^{\text{th}}$  and  $q+1^{\text{th}}$  walking steps, along with the CoM velocity  $v_q^{\text{loc}}$ , is modeled as a function of the footstep length  $u_q^f$  [8].

$$\Delta x_q^{\text{loc}}(u_q^f) = v_q^{\text{loc}} \frac{\sinh(\omega T)}{\omega} + (1 - \cosh(\omega T)) u_q^f, \quad (6)$$

$$v_{q+1}^{\text{loc}}(u_q^f) = \cosh(\omega T) v_q^{\text{loc}} - \omega \sinh(\omega T) u_q^f, \quad (7)$$

where  $\omega = \sqrt{g/H}$  and  $g$  is the gravitational constant and  $H$  is the CoM height. This local dynamics can be transformed into 2D world coordinate by incorporating the robot heading angle  $\phi$  in the world frame:

$$x_{q+1} = x_q + \Delta x_q^{\text{loc}}(u_q^f) \cos(\phi_q), \quad (8)$$

$$y_{q+1} = y_q + \Delta x_q^{\text{loc}}(u_q^f) \sin(\phi_q), \quad (9)$$

$$\phi_{q+1} = \phi_q + u_q^{\Delta\phi}, \quad (10)$$

where  $u_q^{\Delta\phi}$  is the heading angle change during the step  $q$ . For better readability, we express the above dynamics (6)-(10)

TABLE III

NAVIGATION RESULTS FOR *ROUGH TERRAIN* AND *RAMP* ENVIRONMENTS. SUCCESS RATE IS RECORDED BY COUNTING THE NUMBER OF TRIALS THAT SUCCEEDED IN REACHING THE GOAL. MAXIMUM INSTABILITY VALUES ARE TAKEN FROM THE INSTABILITY RECORDING OF EACH SUCCESSFUL TRIAL, AND THEN AVERAGED ACROSS THE TRIALS. PATH SMOOTHNESS IS COMPUTED AS THE MEAN MAGNITUDE OF ANGULAR ACCELERATION WITHIN EACH SUCCESSFUL TRIAL, THEN AVERAGED ACROSS THE TRIALS, WHICH INCREASES WHEN THE ROBOT CHANGES ITS HEADING MORE OFTEN. THE BEST VALUES AMONG THE PLANNERS ARE MARKED IN BOLD, AND ALL OTHER VALUES ARE ANNOTATED WITH THE PERCENTAGE DIFFERENCE RELATIVE TO THE BEST VALUE

Traversability Estimation		STATE (proposed)	LearnedInS		ManualBiped		QuadFoothold	
Global Planner Hyperparameter		$\delta_{\text{limit}} = 3$	$w = 0.5$	$w = 3$	$w = 0.5$	$w = 3$	$w = 0.5$	$w = 3$
Rough terrain	Success Rate	<b>100%</b>	90%	<b>100%</b>	90%	90%	80%	70%
	Max Instability	<b>4.82 (+0%)</b>	7.90 (+64%)	5.37 (+11%)	7.09 (+47%)	8.31 (+72%)	7.50 (+56%)	7.36 (+53%)
	Path Smoothness	<b>0.079 (+0%)</b>	0.119 (+50%)	0.086 (+8.2%)	0.106 (+34%)	0.124 (+56%)	0.114 (+44%)	0.102 (+29%)
	Navigation Time	32.6 s (+33%)	37.3 s (+55%)	36.3 s (+51%)	29.7 s (+23%)	24.3 s (+0.8%)	24.6 s (+2.1%)	<b>24.1 s</b> (+0%)
Ramp	Success Rate	<b>100%</b>	80%	80%	20%	70%	20%	50%
	Max Instability	<b>4.67 (+0%)</b>	7.19 (+54%)	7.39 (+58%)	6.51 (+39%)	5.69 (+21%)	6.89 (+47%)	6.61 (+41%)
	Path Smoothness	<b>0.062 (+0%)</b>	0.150 (+143%)	0.156 (+152%)	0.152 (+146%)	0.115 (+86%)	0.148 (+139%)	0.135 (+118%)
	Navigation Time	38.7 s (+34%)	40.8 s (+41%)	42.6 s (+47%)	39.1 s (+35%)	35.0 s (+21%)	<b>28.9 s</b> (+0%)	29.5 s (+2.1%)

TABLE IV

NAVIGATION RESULTS FOR *CHALLENGING TERRAIN* ENVIRONMENT. EVALUATION METRICS ARE CALCULATED IN THE SAME WAY AS TABLE III

	STATE	LearnedInS	ManualBiped	QuadFoodhold
Success Rate	<b>80%</b>	60%	30%	20%
Max Instability	6.54 (+3.0%)	7.32 (+15%)	<b>6.35</b> (+0%)	12.5 (+97%)
Path Smoothness	<b>0.153</b> (+0%)	0.166 (+8.5%)	0.246 (+61%)	0.164 (+7.2%)
Navigation Time	61.6 s (+24%)	73.5 s (+48%)	255.3 s (+416%)	<b>49.5 s</b> (+0%)

compactly as  $\mathbf{x}_{q+1} = \Phi(\mathbf{x}_q, \mathbf{u}_q)$  with the state variable  $\mathbf{x}_q = (x_q, y_q, \phi_q, v_q^{\text{loc}})$ , and the control variable  $\mathbf{u}_q = (u_q^f, u_q^{\Delta\phi})$ . Using this dynamics model, we formulate our MPC problem as follows:

$$\min_{\mathbf{x}_q, \mathbf{u}_q} m(\mathbf{x}_{N+1}) + \sum_{q=0}^N l(\mathbf{x}_q, \mathbf{u}_q), \quad (11)$$

$$\text{s.t. } \mathbf{x}_0 = \hat{\mathbf{x}}_0, \quad (12)$$

$$\mathbf{x}_{q+1} = \Phi(\mathbf{x}_q, \mathbf{u}_q), \quad (13)$$

$$v_q^{\text{loc}}/v^* + u_q^{\Delta\phi}/(w^*T) \leq 1, \quad (14)$$

where  $\hat{\mathbf{x}}_0$  is the initial state. The cost function includes a terminal cost  $m(\mathbf{x}_{N+1})$  which encourages the final heading  $\phi_{N+1}$  to align with the direction to the local waypoint  $\phi_g$  and the position to reach  $(x_g, y_g)$ , and a running cost  $l(\mathbf{x}_q, \mathbf{u}_q)$  which penalizes excessive velocities to promote smooth motion:

$$m(\mathbf{x}_{N+1}) = w_g \|(x_{N+1}, y_{N+1}) - (x_g, y_g)\|^2 + w_\phi \|\phi_{N+1} - \phi_g\|^2, \quad (15)$$

$$l(\mathbf{x}_q, \mathbf{u}_q) = w_r ((v_q^{\text{loc}})^2 + (u_q^{\Delta\phi})^2). \quad (16)$$

Additionally, the stability-aware command velocity imposes a constraint (14) to regulate command velocities within the stability limit.

## V. EXPERIMENTAL RESULTS

### A. Learning Instability With TravFormer

We collected 72,000 gait cycles data which corresponds to 8 hours simulation time in total for the network training. For evaluation, we tested three ablations: (i) TravFormer without spatial attention, (ii) TravFormer without deformable attention, and (iii) TravFormer with a transformer decoder replaced by an MLP decoder. We report the root mean squared error (RMSE) and the absolute difference of prediction interval coverage probability ( $|\Delta\text{PICP}|$ ) for all samples, with results shown in Table II.

The TravFormer variants with the transformer decoder achieve similarly strong performance, outperforming the MLP decoder by 11.8% in RMSE for instability prediction. Furthermore, the full model—which combines spatial attention with deformable self-attention—achieves the best uncertainty estimation among all variants.

### B. Experimental Setup

1) *Implementation*: We run Digit in MuJoCo simulation with point cloud measurements generated by the simulated sensors. The RRT\* has an iteration number of 500, and the MPC planner has a horizon length  $N$  of 5 walking steps with  $w_g = 1$ ,  $w_\phi = 5$ , and  $w_r = 0.1$ . At the beginning of each test, the robot turns around in place at the starting location for 20 seconds to scan the environment and initialize the elevation map. The elevation map and the traversability of the area are kept updated while the robot explores the environment.

2) *Planner Benchmarking*: To demonstrate the advantages of the proposed method, we design a set of baselines that combine different traversability metrics with various global and local planning strategies. For the cost formulation in global planning, instead of (5), a trade-off between traversability and path length  $c_{ij} = \sum_{k \in M} (1 + w_t(1/t_k)) \Delta l_k$  is used for baselines, where  $t_k$  is the *traversability score* of each approach, and  $w_t$  is the tunable weight [18]. We test two different weight parameters  $w = 0.5$  and 3, to evaluate the effect of the weight on the navigation performance. For local planning, all planners, including the proposed method and the baselines, use the MPC planner introduced

in IV-C. Although the baselines do not define traversability in terms of velocity, we extend their MPC implementations to include a comparable velocity constraint as in (14), ensuring a fair comparison. Specifically, each baseline uses its  $v^*$ ,  $\omega^*$  proportional to its respective traversability score,  $t_k a_{\max}$ , where  $a_{\max} = (0.5 \text{ m/s}, 0.75 \text{ rad/s})$  is the maximum velocity used in Section IV-A. In summary, we test the planners as follows:

- **Stability-Aware Traversability Estimation (STATE):** The planner utilizes the stability-aware command velocity (5), computed with  $\delta_{\text{limit}} = 3$  and  $\alpha = 0.97$ .  $\delta_{\text{limit}} = 3$  is selected since BSFA instability is roughly 2 for stable forward walking at 0.5m/s, so  $\delta_{\text{limit}} = 3$  gives 50% more stability margin for stable walking at 0.5m/s.
- **Learned Instability (LearnedInS):** This method is based on the learned risk-sensitive instability output (1), predicted with a maximum linear command velocity input of  $(0.5 \text{ m/s}, 0)$  and  $\alpha = 0.97$ . We take the scaled inverse of the instability to generate a traversability score  $t_k$ , normalized so that flat terrain has a score of 1, and the higher instability prediction is mapped to a lower score.
- **Manual Traversability Cost Design for Biped (ManualBiped) [6]:** This method uses the manual cost design introduced in [6], which penalizes the slopes and height differences above certain thresholds. Since this design was originally designed to calculate a cost, we apply a scaled inverse transformation in the same manner as in LearnedInS to obtain a traversability score.
- **Quadrupedal Foothold Score (QuadFoothold) [10]:** This method is based on the foothold quality score proposed in [10], originally developed for indicating valid area that quadrupeds can reliably take its foothold on.

3) *Testing Environments:* We validate our approach in various simulated environments. We test the algorithms 10 times each in three 10 m x 10 m MuJoCo environments in Figs. 4–6: *rough terrain*, *ramp*, *challenging terrain*, respectively.

### C. Results

1) *Benefit of Traversability Learning for Bipedal Locomotion:* We first evaluate how well the proposed traversability measures the difficulty in bipedal locomotion in comparison to prior methods. Fig. 4 illustrates the traversability maps and the path plans generated by each planner in *rough terrain* scenario. We observe that the QuadFoothold baseline primarily detects risk along the edges of the gray stepping stones, but fails to capture the overall difficulty of the region. ManualBiped baseline also underestimates the risk posed by the gray stepped regions, likely due to manually chosen thresholds and hand-crafted rules, which might not be fine-tuned to this type of terrain. Consequently, both planners plan paths that traverse the hazardous areas regardless of the traversability cost weight, which results in lower success rates and higher instability in Table. III. In contrast, our STATE method and the LearnedInS baseline with a high traversability weight leverage our bipedal instability-supervised neural network, assessing stepped areas as more dangerous and successfully avoiding them. These approaches achieve a 100% success rate and lower instability records, demonstrating the advantage of a traversability estimator tailored to bipedal stability.

2) *Benefit of Stability-Aware Command Velocity-Based Traversability in Global Planning:* We then examine the robustness of the proposed global planning strategy by comparing the planning results in different environments. Figs. 4 and 5 depict

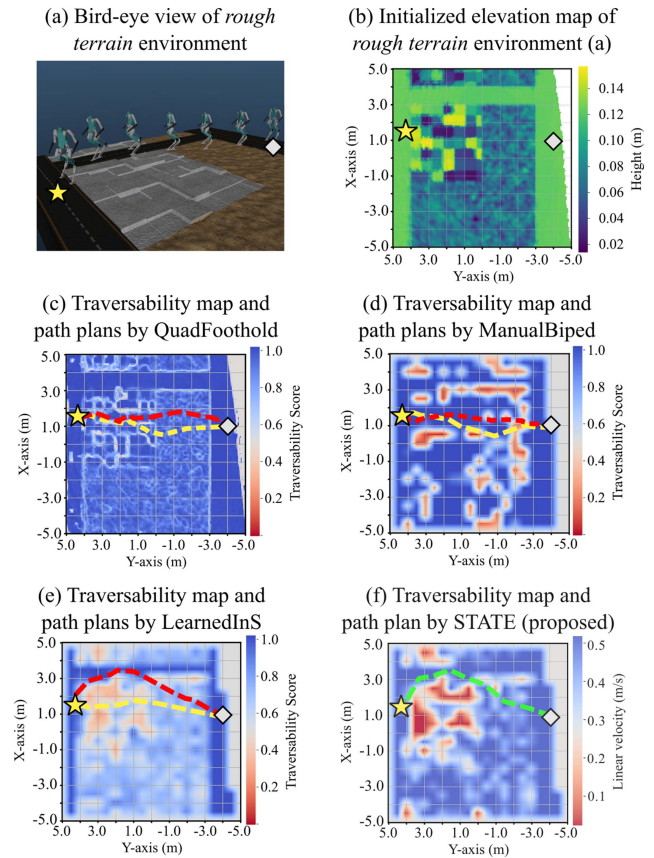


Fig. 4. Comparison of estimated traversability maps and paths planned by the proposed method and baselines. The starting and goal points are marked by a gray diamond and a yellow star, respectively. (a) Diagonal bird-eye view of *rough terrain* environment and the snapshots of the robot traversing along the path by the proposed method. (b) Initialized elevation map of the MuJoCo environment (a). (c), (d), (e) Traversability maps based on (b) and path plans by the QuadFoothold [10], ManualBiped [6], and LearnedInS baselines, respectively. Red and yellow paths are the results of different trade-off weight values  $w = 3$  and 0.5. (f) Traversability map based on (b) and the path plan of the proposed method, STATE, with  $\delta_{\text{limit}} = 3$ .

the path planned by STATE and LearnedInS on *rough terrain* and *ramp* environments, respectively. We can observe that STATE plans a safe and efficient path in both environments with a single hyperparameter  $\delta_{\text{limit}}$ . In contrast, the planning results of LearnedInS are highly dependent on environments. On *rough terrain*, the weight  $w = 3$  produces a safe and time-efficient path, comparable to that of STATE. However, on *ramp*, the same weight  $w = 3$  results in unsafe paths with a lower success rate and higher instability, and even a longer navigation time. This is because the length difference between the shortcut and detour paths is greater in *ramp* than that in *rough terrain*, necessitating a higher weight to trigger a detour. However, our proposed stability-aware command is environment-agnostic, allowing the planner to choose paths that maintain a relatively consistent instability level.

*Remark 2 (Environment-agnostic hyperparameter tuning of STATE):* For the baseline traversability score definition  $t \in [0, 1]$  with the cost  $c_{ij} = \sum_{k \in \mathcal{M}} (1 + w(1/t_k))^p \Delta l_k$ , the hyperparameters  $w$  and  $p$  must be re-tuned for each environment, since the score and path length have unrelated scales and units. As a result, in *ramp* environments, we report that no safe paths are obtained with any  $w$  smaller than 100 under  $p = 1$  (not tested beyond  $w = 100$ ), and stable navigation requires

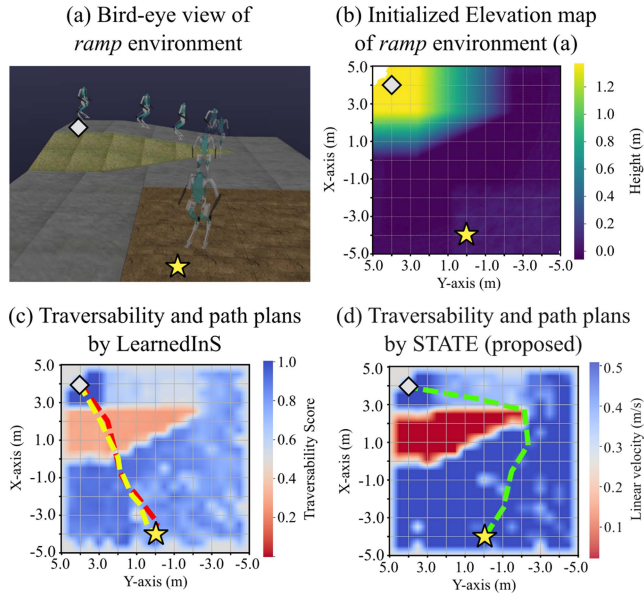


Fig. 5. Comparison of estimated traversability maps and paths planned by the proposed method and baselines. The starting and goal points are marked by a gray diamond and a yellow star, respectively. (a) Diagonal bird-eye view of *ramp* environment and the snapshots of the robot traversing along the path by the proposed method. (b) Initialized elevation map of the MuJoCo environment (a). (c) Traversability map based on (b) and path plans by the LearnedInS baseline. Red and yellow paths are the results of different trade-off weight values  $w = 3$  and  $0.5$ . (d) Traversability map based on (b) and the path plan of the proposed method, STATE, with  $\delta_{limit} = 3$ .

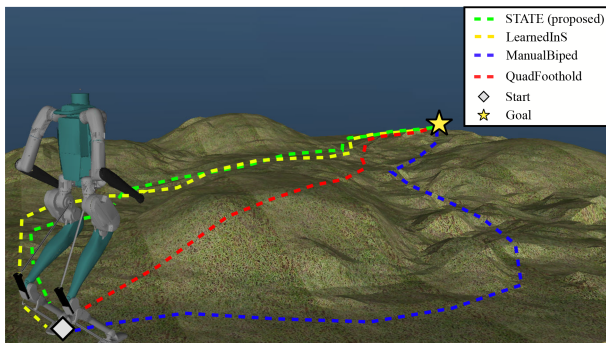


Fig. 6. Path plans of the proposed method and the baselines with the weight  $w = 3$  in *challenging terrain* environment. The starting and goal points are marked by a gray diamond and a yellow star, respectively. Green, yellow, blue, and red paths are by the proposed method (STATE), LearnedInS, ManualBiped, and QuadFoothold planners, respectively.

the introduction of  $p = 2$ . However, with  $p = 2$ , LearnedInS selects an overly conservative path on *rough terrain*, resulting in a longer navigation time of 38.3 s. In contrast, tuning  $\delta_{limit}$  in STATE is intuitive and robot-specific, requiring no iterative tuning across environments. These results highlight the main advantage of STATE: defining traversability by stability-aware velocity enables consistent and environment-agnostic path planning without tedious hyperparameter tuning.

3) *Benefit of Stability-Aware Command Velocity-Based Traversability in MPC Local Planning*: Next, we evaluate how stability-aware command velocity improves the execution of global paths in the local MPC planner. Although a global path may be traversable, it often includes regions requiring delicate locomotion, making terrain-aware velocity regulation critical.

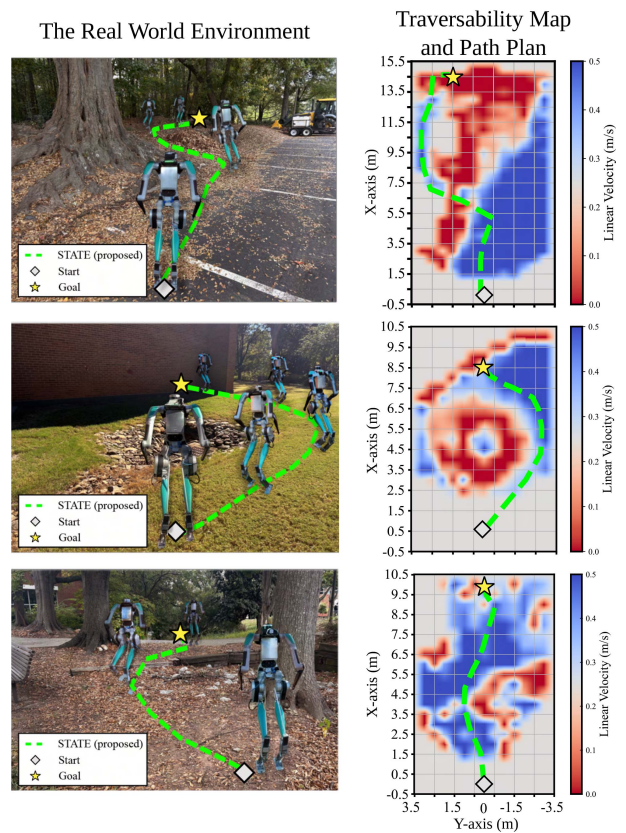


Fig. 7. Real-world demonstration of the proposed navigation stack. The left column presents the testing environments with the robot's executed trajectories overlaid in green. Robot snapshots are extracted from the supplementary video and superimposed for visualization. The right column displays the traversability map and path plan generated by STATE.

To evaluate the differences that arise only from local MPC planning behavior, we use *challenging terrain* environment with a baseline weight setting of  $w = 3$ , where both the proposed method and LearnedInS generate nearly identical global paths but the paths contain slopes and rough terrains that demand the robot careful speed regulation, as shown in Fig. 6. The navigation results are summarized in Table IV. Note that MPC planners without command constraints, commonly used in prior works, consistently failed and are excluded. The proposed algorithm achieves the highest success rate—though not 100%—while maintaining a favorable balance between instability and navigation time shown in Table IV. This improvement is attributed to a more effective adaptation of motion aggressiveness to terrain conditions: accelerating in favorable terrain and modulating speed in unstable terrain. By contrast, the linear scaling in baselines cannot fully capture the nonlinear relationship between terrain, command, and locomotion stability. The full simulation results of all four methods are shown in the supplementary video.

4) *Real-World Validation*: Beyond simulation, we further validate the proposed framework in real-world settings. The robot perceives its environment using a ZED 2i stereo camera mounted on the chest, providing both point cloud perception and localization. All algorithms are run on an external laptop with an Intel Core i7-12700H CPU and an NVIDIA RTX 3060 Mobile GPU. The framework is transferred to hardware with minimal modifications by increasing the MPC horizon length to 7 and reducing the number of TrvRRT\* iterations to 400. The

perception and localization modules operate concurrently, with the traversability map and global plan updated every 5 s, and the MPC local planner running at 3 Hz, ensuring real-time performance. We test the system in three outdoor environments, as shown in Fig. 7. With this navigation stack, TravRRT\* generates safe navigation paths that avoid untraversable regions identified by TravFormer, while the local MPC planner adaptively slows down when crossing ground obstacles or slopes and accelerates on flat terrain to maintain time efficiency. The robot reaches the goal in 54, 32, and 34 s with maximum BSFA instabilities of 3.4, 2.0, and 1.7, respectively, illustrating the practicality of the framework in real world settings. Full experiments, including active velocity modulation, are provided in the supplementary video.

## VI. CONCLUSION AND FUTURE WORKS

This letter presents the first learning-based traversability estimator and navigation framework for bipedal locomotion on diverse rough terrain. Future directions include leveraging the large field of view of humanoids for active exploration of unseen environments. Additionally, incorporating terrain property estimation using semantic perception would be an interesting direction for enhancing robustness.

## REFERENCES

- [1] Z. Gu et al., "Humanoid locomotion and manipulation: Current progress and challenges in control, planning, and learning," *IEEE/ASME Trans. Mechatron.*, 2025.
- [2] E. Krotkov et al., "The DARPA robotics challenge finals: Results and perspectives," in *The DARPA Robotics Challenge Finals: Humanoid Robots to the Rescue*. Berlin, Germany: Springer, 2018, pp. 1–26.
- [3] J.-K. Huang and J. W. Grizzle, "Efficient anytime CLF reactive planning system for a bipedal robot on undulating terrain," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 2093–2110, Jun. 2023.
- [4] S. Kajita et al., "Biped walking pattern generation by using preview control of zero-moment point," in *Proc. 2003 IEEE Int. Conf. Robot. Automat.*, 2003, vol. 2, pp. 1620–1626.
- [5] G. Gibson, O. Dosunmu-Ogunbi, Y. Gong, and J. Grizzle, "Terrain-adaptive, ALIP-based bipedal locomotion controller via model predictive control and virtual constraints," in *Proc. 2022 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 6724–6731.
- [6] S. McCrory, B. Mishra, R. Griffin, J. Pratt, and H. E. Sevil, "Bipedal navigation planning over rough terrain using traversability models," in *Proc. SoutheastCon 2023*, 2023, pp. 89–95.
- [7] K. Muenprasitvej, J. Jiang, A. Shamsah, S. Coogan, and Y. Zhao, "Bipedal safe navigation over uncertain rough terrain: Unifying terrain mapping and locomotion stability," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 11264–11271.
- [8] A. Shamsah, J. Jiang, Z. Yoon, S. Coogan, and Y. Zhao, "Terrain-aware model predictive control of heterogeneous bipedal and aerial robot coordination for search and rescue tasks," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2025, pp. 12352–12358.
- [9] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, "Where should I walk? Predicting terrain properties from images via self-supervised learning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1509–1516, Apr. 2019.
- [10] L. Wellhausen and M. Hutter, "Rough terrain navigation for legged robots using reachability planning and template learning," in *Proc. 2021 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 6914–6921.
- [11] J. Frey, M. Mattamala, N. Chebrou, C. Cadena, M. Fallon, and M. Hutter, "Fast traversability estimation for wild visual navigation," in *Proc. Robot.: Sci. Syst.*, Daegu, Republic of Korea, Jul. 2023.
- [12] X. Cai, M. Everett, L. Sharma, P. R. Osteen, and J. P. How, "Probabilistic traversability model for risk-aware motion planning in off-road environments," in *Proc. 2023 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 11297–11304.
- [13] X. Cai et al., "EVORA: Deep evidential traversability learning for risk-aware off-road autonomy," *IEEE Trans. Robot.*, vol. 40, pp. 3756–3777, 2024.
- [14] M. G. Castro et al., "How does it feel? Self-supervised costmap learning for off-road vehicle traversability," in *Proc. 2023 IEEE Int. Conf. Robot. Automat.*, 2023, pp. 931–938.
- [15] L. Gan, J. W. Grizzle, R. M. Eustice, and M. Ghaffari, "Energy-based legged robots terrain traversability modeling via deep inverse reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 8807–8814, Oct. 2022.
- [16] B. Yang, L. Wellhausen, T. Miki, M. Liu, and M. Hutter, "Real-time optimal navigation planning using learned motion costs," in *Proc. 2021 IEEE Int. Conf. Robot. Automat.*, 2021, pp. 9283–9289.
- [17] A. Dixit, D. D. Fan, K. Otsu, S. Dey, A.-A. Agha-Mohammadi, and J. W. Burdick, "STEP: Stochastic traversability evaluation and planning for risk-aware navigation; results from the Darpa subterranean challenge," *IEEE Trans. Field Robot.*, vol. 2, pp. 81–99, 2025.
- [18] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, "Navigation planning for legged robots in challenging terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 1184–1189.
- [19] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. 2012 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 5026–5033.
- [20] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," *Eng. Appl. Artif. Intell.*, vol. 26, no. 4, pp. 1373–1385, 2013.
- [21] T. Guan, D. Kothandaraman, R. Chandra, A. J. Sathiyamoorthy, K. Weerakoon, and D. Manocha, "GA-Nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 8138–8145, Jul. 2022.
- [22] Y. Lin and D. Berenson, "Humanoid navigation in uneven terrain using learned estimates of traversability," in *Proc. IEEE-RAS 17th Int. Conf. Humanoid Robot.*, 2017, pp. 9–16.
- [23] Y.-C. Lin and D. Berenson, "Long-horizon humanoid navigation planning using traversability estimates and previous experience," *Auton. Robots*, vol. 45, no. 6, pp. 937–956, 2021.
- [24] Z. Li, J. Zeng, S. Chen, and K. Sreenath, "Autonomous navigation of underactuated bipedal robots in height-constrained environments," *Int. J. Robot. Res.*, vol. 42, no. 8, pp. 565–585, 2023.
- [25] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter, "Perceptive locomotion through nonlinear model-predictive control," *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3402–3421, Oct. 2023.
- [26] B. Acosta and M. Posa, "Bipedal walking on constrained footholds with MPC footstep control," in *Proc. IEEE-RAS 22nd Int. Conf. Humanoid Robots*, 2023, pp. 1–8.
- [27] D. McFadden, "Quantitative methods for analysing travel behaviour of individuals: Some recent developments," in *Behavioural Travel Modelling*. Evanston, IL, USA: Routledge, 2021, pp. 279–318.
- [28] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.
- [29] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter, "Elevation mapping for locomotion and navigation using GPU," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 2273–2280.
- [30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
- [31] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
- [32] W. Liu et al., "Tlio: Tight learned inertial odometry," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5653–5660, Oct. 2020.
- [33] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
- [34] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
- [35] W. Chen, R. Khardon, and L. Liu, "Adaptive robotic information gathering via non-stationary Gaussian processes," *Int. J. Robot. Res.*, vol. 43, no. 4, pp. 405–436, 2024.
- [36] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT\*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 2997–3004.