

Continual Learning for Traversability Prediction With Uncertainty-Aware Adaptation

Hojin Lee , Member, IEEE, Yunho Lee , Student Member, IEEE, Daniel A Duecker , Member, IEEE, and Cheolhyeon Kwon , Member, IEEE

Abstract—Traversability prediction is a critical component of autonomous navigation in unstructured environments, where complex and uncertain robot-terrain interactions pose significant challenges such as traction loss and dynamic instability. Despite recent progress in learning-based traversability prediction, these methods often fail to adapt to novel terrains. Even when adaptation is achieved, retaining experience from previously trained environments remains a challenge, a problem known as catastrophic forgetting. To address this challenge, we propose a continual learning framework for traversability prediction that incrementally adapts to new terrains using a generative experience recall model. A key virtue of the proposed framework is two folds: i) retain prior experience without storing past data; and ii) incorporate the uncertainty of the generated samples from the recall model, enabling uncertainty-aware adaptation. Real-world experiments with a skid-steering robot validate the effectiveness of the proposed framework, demonstrating its ability to adapt across a series of diverse environments while mitigating catastrophic forgetting.

Index Terms—Continual learning, planning under uncertainty, field robots, machine learning for robot control.

I. INTRODUCTION

AUTONOMOUS navigation in unstructured environments poses significant challenges due to varying terrain traversability, making traversability prediction a central research focus [1]. Traditionally, traversability prediction has been approached by crafting rules based on terrain geometry [2] and semantic context [3]. However, these methods often fail to capture complex robot-terrain interactions. Recently, learning-based methods have offered promising solutions by leveraging driving data without relying on handcrafted traversability rules [4].

Received 5 June 2025; accepted 22 September 2025. Date of publication 9 October 2025; date of current version 17 October 2025. This article was recommended for publication by Associate Editor M. Benallegue and Editor O. Stasse upon evaluation of the reviewers' comments. This work was supported in part by the National Research Foundation of Korea (NRF) funded by the Korea Government (Ministry of Science and ICT) under Grant RS-2024-00342930 and Grant RS-2020-NR049569 and in part by the InnoCORE Program funded by the Korea Government (Ministry of Science and ICT) under Grant N10250155(00). (Corresponding author: Cheolhyeon Kwon.)

Hojin Lee, Yunho Lee, and Cheolhyeon Kwon are with the Ulsan National Institute of Science and Technology, Ulsan 44919, South Korea (e-mail: hojin-lee@unist.ac.kr; dbs2911@unist.ac.kr; kwonc@unist.ac.kr).

Daniel A Duecker is with the Munich Institute of Robotics and Machine Intelligence (MIRMI), Technical University of Munich (TUM), 80333 München, Germany (e-mail: daniel.duecker@tum.de).

Code available at <https://github.com/HMCL-UNIST/Continual-Traversability-Learning>.

This article has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2025.3619687>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3619687

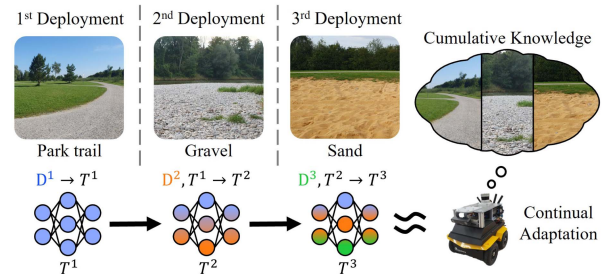


Fig. 1. Continual adaptation across different types of terrains without access to past data. The robot incrementally updates its model while retaining experience for reliable traversability prediction.

While the learning-based traversability prediction methods perform well in conditions similar to the training environment, they often generalize poorly when deployed in novel terrains [5]. To address this limitation, adaptation techniques have been proposed to update traversability models when encountering novel terrain. [6]. However, such adaptation often leads to catastrophic forgetting, where newly acquired experience overwrites prior experience, resulting in degraded performance in previously trained environments.

To facilitate adaptation without catastrophic forgetting, strategies such as regularization-based continual learning and dynamic architectural methods have been explored [7]. Among these methods, experience replay-based continual learning has been widely adopted in traversability learning, as it enables models to update by reusing past data without requiring complex optimization objectives or architectural changes [8], [9]. These methods incrementally expand memory or selectively store data for experience replay. However, they are often hindered by high memory costs and/or the potential loss of valuable information during data selection.

To enable adaptation without losing prior experience and without relying on memory-demanding replay mechanisms, we propose a novel continual learning framework for traversability prediction that features uncertainty-aware adaptation (See Fig. 1.). This framework integrates sensory environmental information and robot state information as input modalities, effectively learning to predict both traversability and its associated uncertainty. A key attribute of the proposed framework is a generative experience recall model, which retrieves prior experience to guide continual updates of both traversability prediction model and itself. In this way, prior experience can be retained without needing memory for storing past data. The generative recall model is further empowered by an uncertainty-aware adaptation

TABLE I
COMPARISON OF LEARNING-BASED TRAVERSABILITY PREDICTION METHODS WITH ADAPTATION CAPABILITY.

	Input modality	Adaptation technique	Prior-experience memory	Uncertainty modeling
[6]	Vision	Adaptation with recent experience	N/A	Epistemic
[9]	Vision	Experience replay	Fixed-sized memory	Aleatoric
[8]	Vision	Experience replay	Incremental dynamic memory	-
Proposed	Vision, geometry, dynamics	Generative experience replay	N/A	Aleatoric, epistemic

mechanism, whereby the uncertainties of recalled samples are evaluated so that trustworthy samples are used to update the recall model. Building on this foundation, the trained traversability prediction model is integrated into a model predictive control (MPC) framework to navigate a robot across a series of diverse environments.

These design choices enable a continual learning framework for traversability prediction that exploits a generative experience recall model for effective (i.e., retention of prior experience), efficient (i.e., no need to store past data), and reliable (i.e., uncertainty-aware) update. Real-world experiments with a skid-steering robot confirm the effectiveness of the proposed framework, showing its adaptability across diverse environments while retaining performance in previously trained environments. The main contributions are as follows:

- We propose an uncertainty-aware traversability prediction method that associates terrain features with the robot dynamics to capture complex robot-terrain interactions.
- We introduce a continual learning framework leveraging generative experience recall and an uncertainty-aware adaptation mechanism to retain prior experience while adapting to novel environments.
- Real-world experiments using a skid-steering robot validate the proposed framework, demonstrating persistent navigation performance while learning uncertain traversability across diverse environments.

II. RELATED WORK

A. Learning-Based Traversability Prediction

Learning-based methods have emerged as a powerful alternative to traditional rule-based methods in traversability prediction. They rely on real or simulated data to model complex robot-terrain interactions with minimal supervision [2]. A critical design choice in prediction models is the input data modality, which directly impacts the model's ability to comprehend underlying dynamics. Geometry features are commonly used [2], but they are insufficient to capture semantic terrain characteristics. To address this, the Learning Applied to Ground Vehicles (LAGR) program exerted pioneering effort on employing visual features for traversability prediction [3].

Learning terrain geometry and visual features may not be sufficient, since traversability is substantially influenced by the robot's state and dynamics. Some methods attempt to exploit proprioceptive sensor data to capture this influence, such as measuring traversability in terms of vibration level [10]. However, they still fall short in fully representing traversability, making it difficult to assess the admissible states for navigating in a given terrain. To comprehend the interdependencies between the robot's state and terrain features in traversability prediction, [11] encompasses the robot's speed, inertial sensor data, and terrain

visual information to establish a forward kinematics model. This model can predict the robot's dynamic response on the terrain, interpreting its traversability. A major limitation of this approach, however, is the lack of uncertainty modeling, which degrades the robustness and reliability of traversability predictions. Furthermore, the aforementioned learning-based methods often fail to generalize when deployed in environments that differ from the training conditions. In response to this shortcoming, we propose a continual learning-based framework that combines terrain features and robot state information for traversability prediction, while also taking into account uncertainty.

B. Adaptation for Traversability Prediction

Learning-based traversability prediction faces challenges in novel environments that differ from the trained environment. Adaptation methods address these challenges by updating the model with newly collected data, focusing on rapid adaptation to new environments [6]. However, they overlook retaining experience from previously trained environments during updates. This leads to degraded navigation performance when the robot is deployed in previous environments.

Experience replay mechanisms have been proposed to address this issue by storing past data in a memory buffer and replaying it alongside newly acquired data [7]. [8] employs an incremental replay buffer that clusters visual features, but its memory footprint grows with environmental diversity, limiting scalability in long-term deployment. In contrast, [9] uses a fixed-size replay buffer, carrying the risk of discarding valuable information when adding new data. These methods present a fundamental trade-off between scalability and representational capacity, making it difficult to retain prior experience across diverse environments. Moreover, neither [8] nor [9] models epistemic uncertainty, which is essential for assessing prediction confidence in unfamiliar environments. They also primarily rely on terrain appearance while neglecting the influence of robot dynamics on traversability, limited in capturing the complexity of robot-terrain interactions.

To address the shortcomings in experience replay mechanisms, generative model-based continual learning offers an alternative by synthesizing past data from learned representations, thereby eliminating the need to store past data and supporting scalable adaptation to diverse environments [7]. While offering notable advantages, such methods often fail to account for the uncertainty associated with generated samples, which undermines the prediction model's updates and potentially compromises performance. To this end, we propose a framework that accounts for the uncertainty of the generated samples and facilitates reliable adaptation to new data and retention of prior experience.

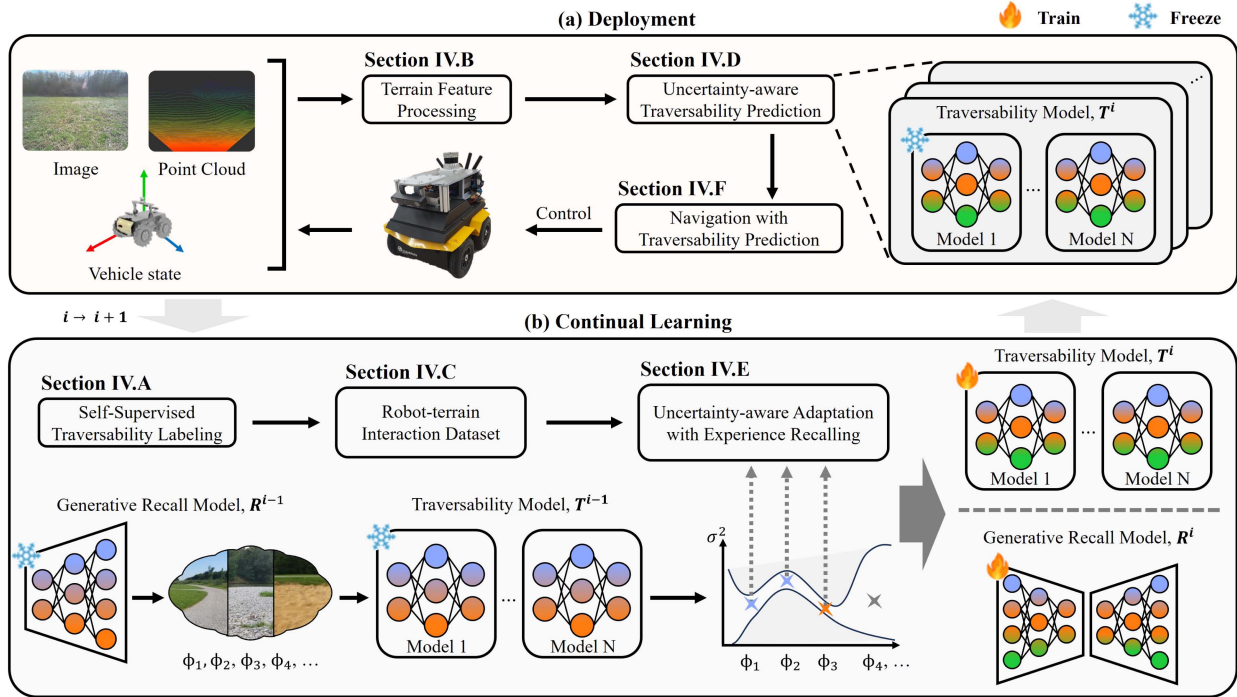


Fig. 2. Overview of the proposed framework. (a) Deployment phase: uncertainty-aware traversability prediction and navigation. (b) Continual learning phase: uncertainty-aware model adaptation to new environments via generative experience recall.

To provide a comparative overview, Table I summarizes key aspects of the representative traversability prediction methods with adaptation, including the proposed framework.

III. PRELIMINARIES

A. Robot Dynamics With Traction Parameters

Let us consider a discrete-time dynamics for a ground robot described by:

$$x_{k+1} = f(x_k, u_k, \xi_k),$$

where x_k represents the robot's state, u_k is the control input, and ξ_k denotes the traction parameter at time step k , accounting for variability in robot-terrain interactions. f is modeled by the following unicycle-based kinematic as it effectively captures the dynamics of a wide range of ground robots, including skid-steering and legged platforms:

$$\begin{bmatrix} p_{k+1}^x \\ p_{k+1}^y \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} p_k^x \\ p_k^y \\ \theta_k \end{bmatrix} + \Delta t \cdot \begin{bmatrix} \xi_k^x \cdot v_k^{cmd} \cdot \cos(\theta_k) \\ \xi_k^y \cdot v_k^{cmd} \cdot \sin(\theta_k) \\ \xi_k^z \cdot \omega_k^{cmd} \end{bmatrix}, \quad (1)$$

where the state vector $x_k = [p_k^x, p_k^y, \theta_k]^T$ represents robot's $X - Y$ position in global frame and its yaw angle. The control input $u_k = [v_k^{cmd}, \omega_k^{cmd}]^T$ consists of the linear and angular velocity commands, and $\Delta t > 0$ denotes the sampling time. The traction parameters $\xi_k = [\xi_k^x, \xi_k^z]^T \in \mathbb{R}^2$ account for the linear and angular discrepancies between the commanded and actual velocities due to uncertain robot-terrain interactions. Without loss of generality, we consider these discrepancies as a measure of traversability [12].

B. Planning With Uncertain Traversability Prediction

In unstructured environments with varying traversability, the navigation problem involves determining the optimal control input sequence $u_{k:k+T-1}^*$ that minimizes the robot's navigation cost over a receding horizon of length T [12]. Given the state x_k , and a navigation cost function \mathcal{C} , the receding horizon stochastic optimal control problem can be formulated as:

$$\begin{aligned} \min_{u_{0:T-1}} \mathbb{E}[\mathcal{C}(x_{0:T}, u_{0:T-1})] \\ \text{s.t. } x_0 = x_k, \\ x_{t+1} = f(x_t, u_t, \hat{\xi}_t), \forall t \in \{0, \dots, T-1\} \end{aligned} \quad (2)$$

where $\hat{\xi}_t \sim \mathcal{P}(\xi|x_t)$ represents the distribution of traction at time step t , which can be predicted by the traversability prediction model. To solve (2), we utilize a sampling-based MPC method, leveraging its parallelizability on GPUs for real-time computation [13].

C. Continual Learning Problem Formulation

To reliably codify $\hat{\xi}$ across different environments, we tackle the problem of domain-incremental continual learning for traversability prediction [7]. We define the traversability prediction model $\mathcal{T} : \mathbb{R}^q \rightarrow \mathbb{R}^2 \times \mathbb{R}_{\geq 0}^2$ as a mapping from input features, comprising the robot's state and terrain features, to the mean and variance of the Gaussian distributions over traction parameters $\hat{\xi} \sim \mathcal{N}(\mu(\cdot), \sigma^2(\cdot))$, where $(\mu(\cdot), \sigma^2(\cdot)) = \mathcal{T}(\cdot)$. The goal is to train a sequence of models $\mathcal{T}^1, \dots, \mathcal{T}^{i-1}$, such that \mathcal{T}^i adapts to new environments encountered during the i^{th} deployment, while retaining experience from previous

($1^{st}, \dots, i-1^{th}$) deployments. In this setting, the input distribution (environmental terrain characteristics) may shift over deployments, whereas the underlying task of learning (traversability prediction model) remains consistent.

At each deployment phase i , the robot collects data \mathcal{D}^i in the corresponding environment, forming a sequential dataset $\mathcal{D}^{1:i} = \{\mathcal{D}^1, \dots, \mathcal{D}^i\}$. Ideally, the model \mathcal{T}^i would be trained on the full dataset $\mathcal{D}^{1:i}$, but storing and accessing all the past data is impractical on resource-constrained platforms. Instead, our continual learning framework aims to update \mathcal{T}^{i-1} using only \mathcal{D}^i , with the goal of producing \mathcal{T}^i that preserves performance without direct access to $\mathcal{D}^{1:i-1}$.

IV. ALGORITHM DEVELOPMENT

This section presents the proposed continual learning framework for traversability prediction, empowered by uncertainty-aware adaptation, as depicted in Fig. 2.

A. Self-Supervised Traversability Labeling

To train the traversability prediction model, we collect samples of traction parameters ξ while navigating over different terrain environments, including both traversable and non-traversable regions. Inspired by the nonlinear moving-horizon estimator in [4], we utilize odometry to estimate the robot's state x and use this information to estimate the traction parameters ξ . Specifically, we employ an Extended Kalman Filter (EKF) where the traction parameters are modeled as a random walk process: $\xi_{k+1} = \xi_k + \nu$. Here, ν is zero-mean Gaussian process noise that accounts for gradual variations over time [14]. Using (1) along with state x_{k+1} , the EKF iteratively refines the traction parameter estimates ξ_k , establishing datasets that are used to train the traversability prediction model.

B. Terrain Feature Processing

We use RGB images and 3D point clouds to build a multi-modal terrain representation based on [15]. The point clouds are processed into an elevation grid map, capturing the geometric properties of the terrain. Simultaneously, visual features are extracted from images using a pre-trained DINOv2 backbone [16], producing pixel-wise feature embeddings. However, projecting these high-dimensional visual features (e.g., 384 dimensions for the DINOv2-small model) into a grid map incurs substantial computational and memory overhead, making it impractical for real-time use on mobile robots.

To address this, we employ an autoencoder network to compress the high-dimensional visual feature embeddings from DINOv2 into a low-dimensional latent feature. The network is trained in an unsupervised manner on the original dense DINOv2 features from a large-scale off-road dataset by optimizing reconstruction loss. Once trained, the encoder part is deployed for real-time processing, receiving dense features from the DINOv2 backbone and generating compact, low-dimensional visual features to be integrated into the multi-modal grid map. As a result, terrain features $\phi \in \mathbb{R}^p$, implicating both geometric and visual information, are assigned to individual grids in the map, as illustrated in Fig. 3. These terrain features are stored in a local grid map centered on the robot's pose, capturing terrain

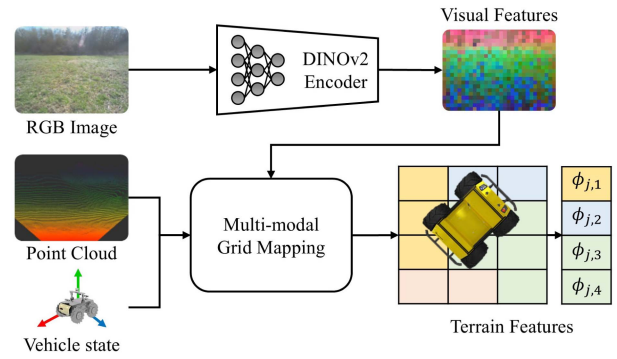


Fig. 3. Grid mapping of the terrain feature ϕ , based on geometric and visual information of the corresponding terrain patches beneath the robot's four wheels.

characteristics over a fixed window for traversability model training and prediction.

C. Robot-Terrain Interaction Dataset

Based on the labeled traction parameter data, driving data, and terrain feature data, we construct a robot-terrain interaction dataset:

$$\mathcal{D} = \{(\phi_j, v_j, \xi_j) \mid j = 1, \dots, \mathcal{B}\},$$

where each sample consists of the terrain feature vector ϕ_j , the robot's velocity state v_j , and the labeled traction parameter ξ_j at time step j . \mathcal{B} denotes the total number of samples in the dataset. The terrain feature vector $\phi_j = [\phi_{j,1}^\top, \phi_{j,2}^\top, \phi_{j,3}^\top, \phi_{j,4}^\top]^\top$ consists of multiple terrain features extracted at different contact points. For instance, in a four-wheeled robot, each $\phi_{j,i}$ corresponds to the terrain features at the i^{th} wheel position ($p_{j,i}^x, p_{j,i}^y$) at time step j . The robot's velocity state $v_j = [v_j^x, w_j^z]^\top$ includes the linear and angular velocities estimated by odometry. To construct the dataset, we retrieve the stored local terrain grid maps for each recorded robot pose. For each map, we project the trajectory history within a local window and extract terrain features at the wheel contact points along the trajectory. These features are then paired with the corresponding robot velocity and traction estimates to form the dataset. Samples with missing terrain features at any wheel contact point due to being outside the sensor field of view are discarded.

D. Uncertainty-Aware Traversability Prediction

Given the dataset \mathcal{D} , the model \mathcal{T} is trained to predict the traction parameter as $\hat{\xi}_k = \mathcal{T}(\phi_k, v_k)$, where ϕ_k represents the terrain features, and v_k denotes the robot velocity state at time step k . To account for uncertainty, we model \mathcal{T} as a probabilistic ensemble model capable of capturing both aleatoric and epistemic uncertainties [17]. Each ensemble member receives the concatenated terrain feature ϕ_k and velocity state v_k , as input and is trained independently. All ensemble members share the same architecture, comprising a multi-layer perceptron with fully connected layers of sizes [16, 32, 32, 16], each followed by a LeakyReLU non-linear activation function. The m^{th} ensemble model \mathcal{T}_m outputs the traction parameter as a Gaussian distribution:

$$\hat{\xi}_{k,m} \sim N(\mu_{k,m}, \sigma_{k,m}^2), \quad \text{where } [\mu_{k,m}, \sigma_{k,m}^2] = \mathcal{T}_m(\phi_k, v_k).$$

Based on predictions from M ensemble models, we compute the final prediction as:

$$\hat{\xi}_k \sim N(\mu_k, \sigma_k^2), \quad \text{where } [\mu_k, \sigma_k^2] = \mathcal{T}(\phi_k, v_k), \quad (3)$$

with the mean and variance computed as:

$$\mu_k = \frac{1}{M} \sum_{m=1}^M \mu_{k,m}, \quad \sigma_k^2 = \sigma_{k,\text{ale}}^2 + \sigma_{k,\text{epi}}^2,$$

where the variances from aleatoric $\sigma_{k,\text{ale}}^2$ and epistemic $\sigma_{k,\text{epi}}^2$ uncertainties are respectively given by:

$$\sigma_{k,\text{ale}}^2 = \frac{1}{M} \sum_{m=1}^M \sigma_{k,m}^2, \quad \sigma_{k,\text{epi}}^2 = \frac{1}{M} \sum_{m=1}^M (\mu_{k,m} - \mu_k)^2.$$

E. Uncertainty-Aware Adaptation With Experience Recalling

To adapt the traversability prediction model across diverse environments, we leverage a generative model-based continual learning framework. The generative model recalls past input-output samples for experience replay without storing raw data. Before delving into the details of our continual learning framework, the initial step is established as follows.

1) *Initial Training of the Traversability Prediction Model:* We begin with the initial model, \mathcal{T}^1 , trained on the first environment dataset \mathcal{D}^1 , which serves as the foundation for adaptation. This initial model \mathcal{T}^1 is trained by minimizing the Negative Log-Likelihood (NLL) loss, denoted as \mathcal{L}_{NLL} , which captures the likelihood of the estimated traction parameters given the corresponding terrain features and the robot's dynamics states, i.e., velocity.

2) *Initial Training of the Generative Experience Recall Model:* We design a generative recall model \mathcal{R}^1 , which randomly synthesizes the prediction model's input and output data, denoted as $\mathcal{D}^{\text{recall}}$. As with the prediction model, the input data consists of two components: terrain features ϕ^{recall} and robot dynamic states, i.e., v^{recall} . Leveraging the fact that velocities are physical quantities with bounded ranges, we design a conditional recall process that first samples velocity and then generates corresponding terrain features conditioned on it. To accomplish this, we adopt a Conditional Variational Autoencoder (CVAE) [18], which consists of an encoder and a decoder, denoted as $q_{\text{enc}}(z|\phi, v)$ and $p_{\text{dec}}(\phi|v, z)$, respectively. Upon training of the CVAE, the decoder is used to recall features ϕ^{recall} conditioned on a sampled velocity state v^{recall} and latent variable z . During recall, both v^{recall} and z are randomly sampled, where v^{recall} respects the velocity limits defined by the robot's hardware specifications, and z is drawn from a Gaussian prior distribution. Next, to synthesize the output data, the trained traversability prediction model \mathcal{T} processes the recalled inputs as follows:

$$\xi^{\text{recall}} \sim \mathcal{N}(\mu^{\text{recall}}, (\sigma^{\text{recall}})^2),$$

where $[\mu^{\text{recall}}, (\sigma^{\text{recall}})^2] = \mathcal{T}^1(\phi^{\text{recall}}, v^{\text{recall}})$. This prediction includes σ^{recall} as a measure of uncertainty, reflecting the model's confidence in its generated samples. Specifically, a higher σ^{recall} indicates that the recalled sample likely originates from an environment that was not well represented in \mathcal{D}^1 for training \mathcal{T}^1 .

3) *Uncertainty-Aware Traversability Prediction Model Update:* We update the traversability prediction model from \mathcal{T}^{i-1} to \mathcal{T}^i for $i \geq 2$, using only the collected dataset \mathcal{D}^i and the

generative recall model \mathcal{R}^{i-1} . First, we generate a batch of recalled input-output samples using \mathcal{R}^{i-1} , overwriting the dataset $\mathcal{D}^{\text{recall}}$. These recalled samples are then combined with the newly collected dataset \mathcal{D}^i to update the model \mathcal{T}^i by minimizing the following loss function:

$$\mathcal{L}_{\mathcal{T}^i} = \mathcal{L}_{\text{NLL}}(\mathcal{D}^i) + \lambda \mathcal{L}_{\text{adapt}}(\mathcal{D}^{\text{recall}}), \quad (4)$$

where λ is a scaling constant. The first term, $\mathcal{L}_{\text{NLL}}(\mathcal{D}^i)$, captures the experience obtained from newly acquired data. The second term, $\mathcal{L}_{\text{adapt}}(\mathcal{D}^{\text{recall}})$, aims to retain prior experience through recalled samples that imitate past data. This is achieved by aligning the distributions of the predicted and recalled output distributions using the Jensen-Shannon (JS) divergence as a distance measure, defined as follows:

$$\begin{aligned} \mathcal{L}_{\text{adapt}}(\mathcal{D}^{\text{recall}}) := & \sum_{\mathcal{D}^{\text{recall}}} \frac{1}{2} KL(\mathcal{T}^i(\phi^{\text{recall}}, v^{\text{recall}}) | \xi^{\text{recall}}) \\ & + \frac{1}{2} KL(\xi^{\text{recall}} | \mathcal{T}^i(\phi^{\text{recall}}, v^{\text{recall}})), \end{aligned}$$

where KL denotes the Kullback-Leibler (KL) divergence. This distributional alignment alleviates catastrophic forgetting of recalled samples and mitigates overfitting to unreliable samples with higher σ^{recall} , thereby reducing the risk of erroneous updates.

4) *Uncertainty-Aware Generative Experience Recall Model Update:* Similar to the prediction model update, we update the generative experience recall model from \mathcal{R}^{i-1} to \mathcal{R}^i using the newly collected dataset \mathcal{D}^i and recalled dataset $\mathcal{D}^{\text{recall}}$, without explicit access to past data, i.e., $\mathcal{D}^{l-1}, \forall l \leq i$. Specifically, we recall data by randomly sampling input features, denoted as $(\phi^{\text{recall}}, v^{\text{recall}})$, using \mathcal{R}^{i-1} . The quality of the recalled samples is assessed using their uncertainty $(\sigma^{\text{recall}})^2$, predicted by the preceding prediction model \mathcal{T}^{i-1} . These uncertainty estimates act as proxies for distributional similarity to real data, enabling us to identify and filter recalled samples that deviate from the real data distribution. To this end, we apply a variance threshold τ to exclude uncertain recalls, yielding the refined recall dataset:

$$\bar{\mathcal{D}}^{\text{recall}} := \{(\phi^{\text{recall}}, v^{\text{recall}}) | \sigma^{\text{recall}} < \tau\}.$$

We then define the augmented dataset as the union of the newly acquired data \mathcal{D}^i and the filtered recalled samples:

$$\mathcal{D}^{\text{aug}} := \mathcal{D}^i \cup \bar{\mathcal{D}}^{\text{recall}}. \quad (5)$$

The generative experience recall model \mathcal{R}^i is trained using \mathcal{D}^{aug} , ensuring that only recalled samples with sufficiently low uncertainty contribute to retaining prior experience while recently acquired data facilitates adaptation to the new environment.

A comprehensive overview of the proposed continual learning framework with uncertainty-aware adaptation is presented in Algorithm 1.

F. Navigation With Traversability Prediction

Based on the updated traversability prediction model, the optimal off-road navigation can be formulated as an MPC problem. The MPC minimizes a cost function associated with predicted future states as outlined in Section III-B. Specifically, the objective cost in (2) is defined as:

$$\mathcal{C}(x_t, \mu_t, \sigma_t^2) = w_1 \text{Goal}(p_t^x, p_t^y) + w_2 \sigma_t^2 + w_3 |1 - \mu_t|,$$

Algorithm 1: Continual Learning Framework With Uncertainty-Aware Adaptation.

Initialize: Set iteration index $i = 1$.

Input: Initial dataset \mathcal{D}^1 .

- 1: Initial train \mathcal{T}^1 and \mathcal{R}^1 (Sections IV.E.1 and IV.E.2).
 - 2: **while** New dataset \mathcal{D}^{i+1} is acquired **do**
 - 3: Generate recalled dataset $\mathcal{D}^{\text{recall}}$ using \mathcal{R}^i and \mathcal{T}^i .
 - 4: Update $\mathcal{T}^i \rightarrow \mathcal{T}^{i+1}$ with (4) (Section IV.E.3).
 - 5: Update $\mathcal{R}^i \rightarrow \mathcal{R}^{i+1}$ using \mathcal{D}^{aug} in (5) (Section IV.E.4).
 - 6: $i = i + 1$.
 - 7: **end while**
-

where $\text{Goal}(p_t^x, p_t^y)$ encourages progress toward the goal over a finite horizon, and σ_t^2 penalizes regions of high predictive uncertainty stemming from limited environment knowledge and/or irreducible measurement noise. The slip cost term $|1 - \mu_t|$ encourages the predicted traction to approach 1, representing ideal alignment between the robot's actual motion and the commanded input under the nominal dynamics. Notably, μ_t is not constrained to lie within $[0, 1]$, allowing the model to capture a wider range of robot-terrain interactions. The weights $w_1, w_2, w_3, \in \mathbb{R}^+$ are tunable scaling factors.

To solve for (2), which is subject to stochastic dynamics f , we adopt a mean-propagation strategy for cost approximation, following the method in [17]. Specifically, the robot dynamics are propagated using the mean estimates of uncertain traction parameters. The propagated states are then used to evaluate the cost function and optimize the control inputs. Under this strategy, the navigation problem can be expressed as the following MPC problem:

$$\begin{aligned} \min_{u_{0:T-1}} \quad & \sum_{t=0}^T \mathcal{C}(x_t, \mu_t, \sigma_t^2) \\ \text{s.t.} \quad & x_0 = x_k, \\ & x_{t+1} = f(x_t, u_t, \mu_t), \forall t \in \{0, \dots, T-1\} \\ & [\mu_t, \sigma_t^2] = \mathcal{T}(\phi_t, v_t), \forall t \in \{0, \dots, T-1\}. \end{aligned}$$

Upon optimization, only the first control u_0 is applied to the robot, while the remaining $u_{1:T-1}$ are temporarily stored in a buffer and used as fallback actions if the subsequent MPC computation fails to meet the desired control frequency.

V. FIELD EXPERIMENT

A. Platform and Dataset Description

The experiments are conducted on a Clearpath Jackal robot, equipped with an NVIDIA Jetson AGX Orin for onboard computation, an Ouster LiDAR, and a ZED X stereo camera to sense its surrounding environment. A Lidar-inertial odometry [19] provides high-frequency pose and velocity estimates. The ZED X camera captures both RGB images and depth point clouds, which are used to extract the terrain features. The extracted features are encoded into a multi-modal grid map with a resolution of 0.25 meters [15], and represented as latent vectors of dimension $p = 3$, updated at a rate of 20 Hz. An ensemble of $M = 5$ models

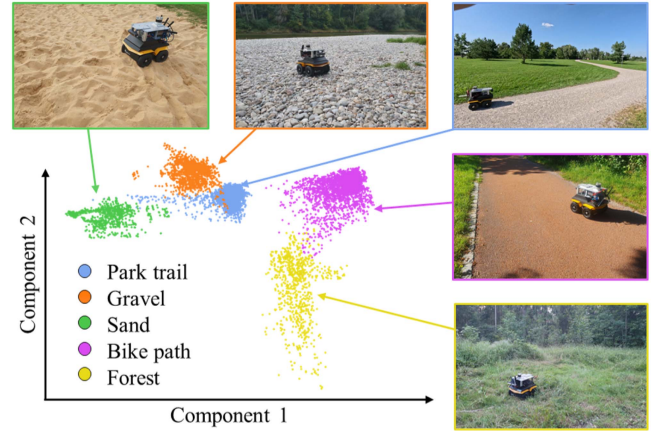


Fig. 4. PCA projection of terrain features ϕ from five distinct environments.

performs traversability prediction. The entire system is built within the ROS 2 environment. The control horizon is set to $T = 30$ with a time step of $\Delta t = 0.1$ seconds, corresponding to a 3-second prediction window. A batch of 1024 rollouts is used per optimization cycle to support a control frequency of 10 Hz.

The evaluation of continual learning necessitates diverse environments, which enables assessment of how well prior knowledge is retained when adapting to novel environments. In light of this requirement, we consider five environments with distinct terrain characteristics: an asphalt trail in a public park (Dataset \mathcal{D}^1), a gravel terrain near a river shore (Dataset \mathcal{D}^2), a sand area (Dataset \mathcal{D}^3), a bike path (Dataset \mathcal{D}^4), and a forest with dense vegetation (Dataset \mathcal{D}^5). At each site, we collected approximately 10 minutes of data by manually driving through both traversable and non-traversable areas. We apply Principal Component Analysis (PCA) to the extracted terrain features ϕ , projecting them into a two-dimensional space to characterize distributional differences among the five environments, as illustrated in Fig. 4. The projected features reveal distinct domain shifts that are pertinent to a domain-incremental continual learning problem, as discussed in Section III-C.

B. Evaluation Metrics

The traversability prediction model is trained sequentially on datasets \mathcal{D}^1 through \mathcal{D}^5 , each corresponding to data collected from Environments 1 through 5. Continual learning performance is evaluated in terms of *adaptation* and *memory retention*. Both are computed from the prediction performance of the traction model on test datasets using the Negative Log-Likelihood (NLL). Let $n_{k,j} \in \mathbb{R}$, $k \leq j$ denote the NLL on the test set of \mathcal{D}^k after training on datasets $\mathcal{D}^{1:j}$. Adaptation performance at the environment j is given by $n_{j,j}$, which measures how well the updated model \mathcal{T}^j fits \mathcal{D}^j . A lower $n_{j,j}$ indicates better adaptation performance. Memory retention performance is evaluated using the Forgetting Measure (FM) [7], defined as $f_{k,j} = \max(0, n_{k,j} - \min_{i < j} n_{k,i})$, which captures the maximum performance degradation on \mathcal{D}^k after training on \mathcal{D}^j . A lower FM indicates better experience retention and improved continual learning performance.

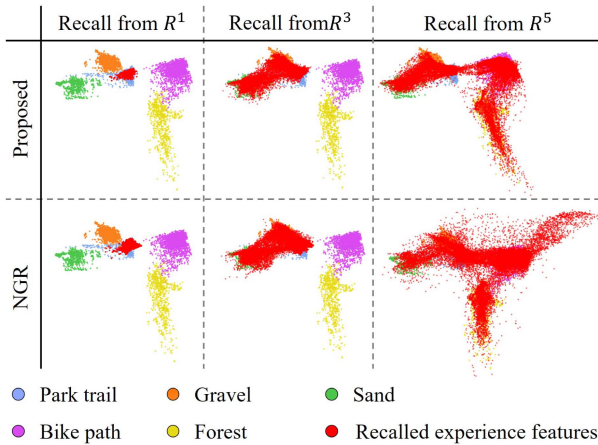


Fig. 5. PCA projection of recalled samples alongside real terrain features collected from Environments 1–5.

C. Baseline Methods

For comparative analysis, we evaluate the proposed continual learning framework against five baseline methods:

- *Complete-Memory (CM)*: This upper-bound baseline trains \mathcal{T}^i using the full dataset \mathcal{D}^j , $1 \leq j \leq i$, enabling batch retraining with complete access to past data.
- *Incremental-Memory (IMOST)* [8]: This method updates \mathcal{T}^{i-1} using an incremental dynamic memory buffer that stores new samples according to an information-expansion criterion.
- *Fast Adaptation (WVN)* [6]: This baseline updates \mathcal{T}^{i-1} using only the recently acquired data \mathcal{D}^i , without considering experience retention.
- *LwF-fashion (LwF)* [20]: This method uses the model \mathcal{T}^{i-1} and the recently acquired data \mathcal{D}^i to retrain \mathcal{T}^i without accessing past data. Experience retention is achieved via distillation by minimizing the KL divergence between the outputs of \mathcal{T}^{i-1} and \mathcal{T}^i on \mathcal{D}^i .
- *Naive Generative Rehearsal (NGR)*: A variant of our method that updates \mathcal{T}^{i-1} with recalled samples \mathcal{R}^{i-1} and the recently acquired data \mathcal{D}^i without the uncertainty-aware adaptation in Section IV.E.4 IV-E4, serving as an ablation of the threshold τ .

To ensure a fair comparison, all baseline methods are interfaced to take terrain and robot states as inputs and to output probabilistic traction, as in the proposed framework. For each method, we identify the hyperparameter setting on the Pareto front with equal weighting of adaptation and memory retention.

D. Field Experiment Results and Discussion

1) *Evaluation of Experience Recalling*: We begin by assessing the validity of recalled samples generated by the generative experience recall model \mathcal{R}^i during adaptation. The distributions of the recalled terrain features are visualized on top of the real terrain features using PCA, as shown in Fig. 5. While the proposed method produces samples that closely align with the real data distribution, the ablation method (NGR) generates dispersed samples that poorly align with real data. These results highlight that the uncertainty-aware adaptation introduced in

TABLE II
ADAPTATION STATISTICS DURING CONTINUAL LEARNING (MEAN \pm STD).

NLL, $n_{j,j}$		Training from \mathcal{D}^1 to \mathcal{D}^j			
		$\mathcal{D}^1 \rightarrow \mathcal{D}^2$	$\mathcal{D}^1 \rightarrow \mathcal{D}^2 \rightarrow \mathcal{D}^3$	$\mathcal{D}^1 \rightarrow \dots \rightarrow \mathcal{D}^4$	$\mathcal{D}^1 \rightarrow \dots \rightarrow \mathcal{D}^5$
Test j		\mathcal{D}^2	\mathcal{D}^3	\mathcal{D}^4	\mathcal{D}^5
Method	CM	0.17 \pm 0.15	-0.61 \pm 0.09	0.89 \pm 0.13	-0.66 \pm 0.06
	IMOST	0.50 \pm 0.26	-0.46 \pm 0.08	1.27 \pm 0.07	-0.35 \pm 0.06
	WVN	0.13 \pm 0.08	-0.57 \pm 0.07	1.06 \pm 0.01	-0.44 \pm 0.07
	LwF	0.19 \pm 0.12	-0.55 \pm 0.07	1.03 \pm 0.11	-0.57 \pm 0.06
	NGR	0.82 \pm 0.34	-0.49 \pm 0.06	1.20 \pm 0.21	-0.43 \pm 0.02
Proposed	0.71 \pm 0.29	-0.25 \pm 0.11	0.97 \pm 0.30	-0.47 \pm 0.05	

TABLE III
FM STATISTICS DURING CONTINUAL LEARNING (MEAN \pm STD).

FM, $f_{k,j}$		Training from \mathcal{D}^1 to \mathcal{D}^j			
		$\mathcal{D}^1 \rightarrow \mathcal{D}^2$	$\mathcal{D}^1 \rightarrow \mathcal{D}^2 \rightarrow \mathcal{D}^3$	$\mathcal{D}^1 \rightarrow \dots \rightarrow \mathcal{D}^4$	$\mathcal{D}^1 \rightarrow \dots \rightarrow \mathcal{D}^5$
Test k	Method				
\mathcal{D}^1	CM	0.60 \pm 0.29	0.57 \pm 0.12	0.70 \pm 0.20	0.76 \pm 0.17
	IMOST	0.73 \pm 0.10	1.12 \pm 0.12	1.33 \pm 0.11	1.55 \pm 0.24
	WVN	1.53 \pm 0.01	2.08 \pm 0.03	1.95 \pm 0.11	2.56 \pm 0.03
	LwF	1.60 \pm 0.12	2.24 \pm 0.23	2.06 \pm 0.14	2.68 \pm 0.16
	NGR	0.80 \pm 0.05	1.38 \pm 0.08	1.42 \pm 0.06	1.82 \pm 0.21
Proposed	0.67 \pm 0.05	1.11 \pm 0.07	1.24 \pm 0.08	1.45 \pm 0.13	
\mathcal{D}^2	CM	–	0.12 \pm 0.12	0.20 \pm 0.14	0.22 \pm 0.09
	IMOST	–	0.38 \pm 0.25	0.67 \pm 0.19	1.03 \pm 0.51
	WVN	–	2.17 \pm 0.32	2.28 \pm 0.39	2.29 \pm 0.19
	LwF	–	2.33 \pm 0.34	2.26 \pm 0.30	2.72 \pm 0.18
	NGR	–	0.55 \pm 0.15	0.68 \pm 0.14	1.22 \pm 0.37
Proposed	–	0.38 \pm 0.27	0.68 \pm 0.18	0.98 \pm 0.23	
\mathcal{D}^3	CM	–	–	0.35 \pm 0.17	0.49 \pm 0.20
	IMOST	–	–	1.24 \pm 0.23	1.51 \pm 0.22
	WVN	–	–	3.07 \pm 0.33	3.18 \pm 0.25
	LwF	–	–	2.85 \pm 0.11	3.50 \pm 0.45
	NGR	–	–	1.39 \pm 0.14	2.13 \pm 0.19
Proposed	–	–	0.93 \pm 0.21	1.49 \pm 0.31	
\mathcal{D}^4	CM	–	–	–	0.17 \pm 0.12
	IMOST	–	–	–	0.36 \pm 0.15
	WVN	–	–	–	1.60 \pm 0.21
	LwF	–	–	–	1.52 \pm 0.17
	NGR	–	–	–	0.29 \pm 0.11
Proposed	–	–	–	0.25 \pm 0.16	

Section IV.E.4 IV.E.4 helps the recall model produce samples that better reflect collected experience, supporting more accurate model updates during continual learning.

2) *Evaluation of Adaptation Performance*: As part of the quantitative evaluation, Tables II and III report the mean and standard deviation of NLL and FM over ten training sessions for each continual learning scenario. The CM method consistently achieves the lowest forgetting, as it keeps access to the full dataset throughout training. Among those without direct access to past data, our proposed method yields the lowest FM, indicating improved retention of prior experience while effectively adapting to new environments. The IMOST method achieves retention performance comparable to, and occasionally exceeding, our method, but this comes at the cost of a continually growing memory requirement. This highlights the effectiveness of our uncertainty-aware adaptation for continual learning without storing past data, retaining prior experience with only the parameters of the prediction and recall models.

Apart from the quantitative findings, we qualitatively evaluate the traversability prediction model’s ability to retain prior experience through navigation trials. Specifically, we assess the performance of model \mathcal{T}^5 in Environments 1 after sequential

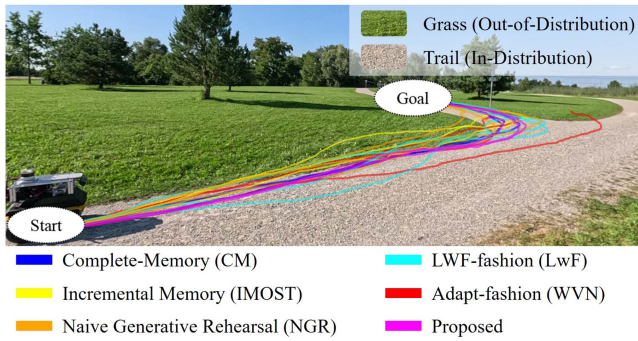


Fig. 6. Navigation results in Environment 1 using the traversability prediction model \mathcal{T}^5 , with the paths of each method overlaid on the field images.

TABLE IV
NAVIGATION RESULTS ACROSS 3 TRIALS PER METHOD. VALUES INDICATE THE NUMBER OF SUCCESSFUL TRIALS.

Metric	Method					
	CM	IMOST	WVN	LwF	NGR	Proposed
Goal reached	3	3	2	3	3	3
OoD avoidance	3	0	0	0	1	3

training on Environments 1 through 5 (i.e., $\mathcal{D}^1 \rightarrow \dots \rightarrow \mathcal{D}^5$). To probe the impact of uncertainty-aware navigation under continual learning, the grass region in the park (treated as an OoD area) was excluded from data collection, while only the asphalt region (in-distribution) was traversed during training. Fig. 6 illustrates representative trajectories from three trials per method, and Table IV summarizes the navigation results. Here, Goal reached denotes the number of successful goal completions out of three trials, and OoD avoidance indicates the number of trials in which the robot successfully avoided entering the grass region. The WVN method fails to reach the goal in one trial, reflecting degraded predictions due to forgetting. Other baselines, except CM and the proposed method, also fail to retain prior experience and incorrectly estimate uncertainty in the OoD region, resulting in poor avoidance performance. In contrast, the proposed method demonstrates stronger experience retention, reliably reaching the goal while avoiding OoD areas. Despite being trained without access to past data, it performs comparably to the CM method. These results demonstrate that our method effectively retains prior experience, enabling reliable navigation under continual learning scenarios.

VI. CONCLUSION

This letter presents a novel continual learning framework for traversability prediction in unstructured environments, addressing the challenges of adaptation and catastrophic forgetting without requiring explicit storage of past data. The core contribution is an uncertainty-aware adaptation strategy that selectively updates both the traversability prediction model and the generative experience recall model based on uncertainty estimates of recalled experiences. Real-world experiments with a skid-steering robot validate the effectiveness of the proposed method, demonstrating improved experience retention and reli-

able navigation performance. Future work will investigate more generalizable learning strategies that leverage prior knowledge, such as physics-informed foundation models, for traversability prediction.

REFERENCES

- [1] C. Sevastopoulos and S. Konstantopoulos, "A survey of traversability estimation for mobile robots," *IEEE Access*, vol. 10, pp. 96331–96347, 2022.
- [2] J. Ahtiainen, T. Stoyanov, and J. Saarinen, "Normal distributions transform traversability maps: LiDAR-only approach for traversability mapping in outdoor environments," *J. Field Robot.*, vol. 34, no. 3, pp. 600–621, 2017.
- [3] D. Kim, J. Sun, S. M. Oh, J. M. Rehg, and A. F. Bobick, "Traversability classification using unsupervised on-line visual learning for outdoor robot navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2006, pp. 518–525.
- [4] M. V. Gasparino et al., "WayFAST: Navigation with predictive traversability in the field," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 10651–10658, Oct. 2022.
- [5] X. Cai et al., "PIETRA: Physics-informed evidential learning for traversing out-of-distribution terrain," *IEEE Robot. Automat. Lett.*, vol. 10, no. 3, pp. 2359–2366, Mar. 2025.
- [6] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter, "Fast traversability estimation for wild visual navigation," in *Proc. Robot.: Sci. Syst.*, Daegu, Republic of Korea, 2023.
- [7] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5362–5383, Aug. 2024.
- [8] K. Ma, Z. Sun, C. Xiong, Q. Zhu, K. Wang, and L. Pei, "IMOST: Incremental memory mechanism with online self-supervision for continual traversability learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2025, pp. 8788–8794.
- [9] H.-S. Yoon, J.-H. Hwang, C. Kim, E. I. Son, S.-W. Yoo, and S.-W. Seo, "Adaptive robot traversability estimation based on self-supervised online continual learning in unstructured environments," *IEEE Robot. Automat. Lett.*, vol. 9, no. 6, pp. 4902–4909, Jun. 2024.
- [10] 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. Automat. Lett.*, vol. 4, no. 2, pp. 1509–1516, Apr. 2019.
- [11] A. Pokhrel, A. Datar, M. Nazeri, and X. Xiao, "CAHSOR: Competence-Aware High-Speed Off-Road Ground Navigation in $\mathbb{S}^2 \times \mathbb{R}^2$," *IEEE Robot. Automat. Lett.*, vol. 9, no. 11, pp. 9653–9660, Nov. 2024, doi: 10.1109/LRA.2024.3457369.
- [12] X. Cai et al., "EVORA: Deep evidential traversability learning for risk-aware off-road autonomy," *IEEE Trans. Robot.*, vol. 40, pp. 3756–3777, 2024.
- [13] H. Lee, J. Kwon, and C. Kwon, "Learning-based uncertainty-aware navigation in 3D off-road terrains," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 10061–10068.
- [14] M. Bloesch et al., "State estimation for legged robots: Consistent fusion of leg kinematics and IMU," in *Robotics: Science and Systems VIII*, MIT Press, 2013, pp. 17–24.
- [15] G. Erni, J. Frey, T. Miki, M. Mattamala, and M. Hutter, "MEM: Multi-modal elevation mapping for robotics and learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 11011–11018.
- [16] M. Oquab et al., "DINOv2: Learning robust visual features without supervision," *Trans. Mach. Learn. Res.*, 2024.
- [17] H. Lee, T. Kim, J. Mun, and W. Lee, "Learning terrain-aware kinodynamic model for autonomous off-road rally driving with model predictive path integral control," *IEEE Robot. Automat. Lett.*, vol. 8, no. 11, pp. 7663–7670, Nov. 2023.
- [18] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3483–3491.
- [19] K. Chen, R. Nemiroff, and B. T. Lopez, "Direct LiDAR-inertial odometry: Lightweight LIO with continuous-time motion correction," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 3983–3989.
- [20] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.