2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

RIDER: Reinforcement-Based Inferred Dynamics via Emulating Rehearsals for Robot Navigation in Unstructured Environments

Sriram Siva and Maggie Wigness

Abstract— **Autonomous navigation in unstructured environments is a challenging task due to the complex and dynamic nature of robot-terrain interactions. Existing approaches often struggle to generalize amidst the complexities of real-world settings. They tend to rely on hand-engineered, rule-based robot models or static weightings assigned to obstacles, semantics, and other perceptual cues to estimate traversability. To address these challenges, we propose a novel approach called Reinforcement-Based Inferred Dynamics via Emulating Rehearsals (RIDER), that learns the dynamics of robot-terrain interactions within a compact latent space, capturing robot's traversability. Operating within a reinforcement learning paradigm, RIDER learns to infer its own dynamics by predicting how future robot observations and states evolve within this latent space in response to navigational behaviors. Furthermore, our approach leverages emulated rehearsals, where the robot learns within the latent space to predict its rewards and generate navigational behaviors, even when real observations have not been updated. Accordingly, RIDER equips robots with the ability to generate navigational behaviors by predicting environmental changes, and plan beyond the speed at which observations from sensors are available. Experimental results and comparisons with baseline methods establish that our proposed method outperforms other approaches in cluttered and unstructured environments and demonstrates an enhanced capacity for autonomous navigation in real-world settings.**

Fig. 1. In unstructured environments, autonomous mobile robots must predict how their behaviors will impact terrain observations and robot states to navigate effectively. Understanding robot-terrain interactions enables them to develop safer and more efficient navigational behaviors for challenging environments with diverse terrain characteristics and obstacles.

I. Introduction

Autonomous mobile robots have become indispensable across many real-world applications involving navigation in unstructured and complex environments. Critical domains such as disaster response, search and rescue, subterranean exploration, and planetary surveys rely increasingly on capable and adaptable robot navigation [1], [2], [3], [4]. However, these off-road environments pose significant challenges due to their unpredictable and dynamic nature. As robots encounter changing terrain conditions, the capacity to consistently anticipate environmental interactions and adjust navigational strategies becomes pivotal for their successful traversal and efficient operation within these dynamic field environments, as depicted in Figure 1. Learning intelligent navigation behaviors that incorporate knowledge of robot-terrain dynamics and how robot states change with terrain will enable autonomous robots to fully utilize their potential to navigate these environments.

Over the years, various methods have been introduced to address the challenge of robot navigation in unstructured terrains. These methods encompass a range of approaches, including rule-based systems for modeling a robot's dynamics [5], [6], [7], [8], supervised learning techniques that imitate navigation behaviors through human demonstrations [9], [10], [11], online learning methods [12], [10], [13] for continuous adaptation, and reinforcement learning (RL) paradigms [14], [15], [16] for iterative learning. More recently, methods have been proposed that leverage machine learning-based control that integrates data-driven formulations to improve pre-established robot models for navigation [17], [18], [19], [20]. Such approaches offer the potential for robots to become more adaptable and sample-efficient learners.

Despite recent advancements, existing robot navigation methods still face several key limitations in unstructured environments. First, planning adjustments are limited by the frequency of sensor updates, which does not provide the rapid adaptation needed in unpredictable environments. Second, reliance on predefined static weightings for obstacles, semantics, and other perceptual cues to estimate traversability limits their ability to generalize to novel terrains or navigational scenarios. Finally, separate modeling of environmental terrain and the robot itself overlooks the critical interactions between them, a gap that becomes increasingly problematic as terrain characteristics change frequently. In addition to these limitations, there are also opportunities to more effectively utilize multi-sensory data to better capture how robot navigational behaviors affect their states and trajectories, especially in environments where terrain dynamically changes as a robot maneuvers. We hypothesize that addressing robot navigation directly at the intersection of planning and perception learning

*Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-23-2-0169.

Sriram Siva and Maggie Wigness are with the US Army DEVCOM Army Research Laboratory, Adelphi, MD 20783, USA. Email: sivasriram2112@gmail.com, maggie.b.wigness.civ@army.mil

979-8-3503-8457-4/24/$31.00 ©2024 IEEE

will be crucial to enable more robust robot navigation in complex real-world terrains.

To address these persistent limitations, we introduce a novel approach called *Reinforcement-Based Inferred Dynamics via Emulating Rehearsals* (RIDER). Our approach learns to encode robot-terrain interactions within a low-dimensional traversability representation space. This enables, for the first time, real-time navigation without reliance on fixed weighting between multi-sensor observations. RIDER then learns to predict future traversability representations based on a robot's navigational behaviors, capturing future environmental changes and robot states. Additionally, within a reinforcement learning framework, RIDER leverages emulated rehearsals, where a robot simulates experiences by predicting rewards and planning behaviors using the learned traversability representations, without needing new real-world observations. This allows a robot to learn navigational behaviors efficiently without complete dependence on real-world robot observation data. Altogether, within a unified learning framework, RIDER equips robots with the ability to predict environmental changes, predict future robot states, and execute intelligent navigational behaviors to traverse through unstructured terrains efficiently.

II. Related Work

A. Classical Control Based Methods

Methods within the domain of classical control theory use predefined models to generate robust navigational behaviors in outdoor field environments. Early approaches included fuzzy logic implementations, lacking prior knowledge of robot dynamics but facilitating navigation [21], [22]. Subsequently, system identification-based techniques emerged, enabling the learning of robot dynamics for improved navigation [5], [23]. Methods such as dynamic programming [24], [25], relied on presumed environmental models or linear quadratic regulators (LQR) [26], [27], to learn nonlinear robot dynamics models for guiding navigation to goal positions. Recent advancements introduced closed-loop feedback control through techniques like model predictive control [28], [7] and model predictive path integral control [6]. These approaches demonstrate the capability to generate resilient navigational behaviors for robots in unstructured terrains characterized by uncertainties.

However, these methods underutilize valuable multi-sensory terrain data, lacking a comprehensive understanding of the terrain, which hinders adaptability and often results in navigation failures or suboptimal traversal, especially in uncertain terrains.

B. Learning Based Methods

Learning-based methods leverage data-driven formulations to generate navigational behaviors across diverse environments. Early approaches for nonlinear robot systems employed high-dimensional linear observations [29], [30]. Subsequently, learning from demonstration (LfD) techniques facilitated the transfer of expert demonstrations to ground robots [9], [10], [11], [31]. Most recent methods on robot navigation tightly integrate robot perception with navigation. For example, terrain classification [9], [32], [33], localization and mapping [34], [35], [36], object detection and tracking [37], [38], [39], and perceptual navigation [40], [41], [42], [43], all have been successfully used to leverage multi-sensory terrain observations for learning to navigation in unstructured environments. Additionally, semantic segmentation-based methods have emerged in this domain, classifying terrains and images based on their navigability for robots [32], [44]. Recent advancements involve modeling the increasing complexity of terrain traversal by learning visual features corresponding to color and texture variations in different terrains [45].

However, these methods often rely on human demonstrations or annotations, which limits adaptability and results in overly conservative behavior. In addition, they often lack the ability to anticipate how their states and environmental observations will change with varying terrains and plan accordingly.

C. Machine Learning Control Based Methods

Machine learning-based control methods learn to generate navigational behaviors by combining data-driven formulations into predefined robot models [17], [18], [19], [20]. Earlier methods used Dynamics Mode Decomposition (DMD) [46], [47] and Sparse Identification of Non-Linear Dynamics (SINDy) [48], [49] to learn data-driven models based on system identification and performed terrain navigation [50], [51]. Evolutionary algorithms were later developed to optimize parameters of a robot model in an online learning fashion for robust navigation [52], [53]. For robots with multiple degrees of freedom, methods were developed that use a combination of iterative Linear Quadratic Regulators (iLQR) and machine learning search to explore multiple robot configurations and plan self-adaptive navigation [54], [55].

Existing methods for robot navigation in unstructured environments rely on pre-fixed weighing of robot observations to determine traversability, which can be inefficient and difficult to model all terrain maneuver capabilities. They also face challenges such as slow convergence and lack of adaptability to diverse environments.

III. Approach

In this section, we present our RIDER approach to address the challenges of navigation in unstructured environments.

A. Inferring Robot Terrain Interaction Dynamics via Learning Traversability Representations

Our approach learns a low dimensional traversability representation space for navigation, where traversability is encoded using robot states and observations. Unlike representations that define fixed-weights for multiple observation types (e.g., obstacles, semantics, etc.) or use the entire observation space, we estimate traversability from multisensory data, which allows us to learn what is relevant from the high-dimensional multisensory observations and discard the rest. Instead of directly predicting future robot states or observations, we forecast future traversability representations, which helps

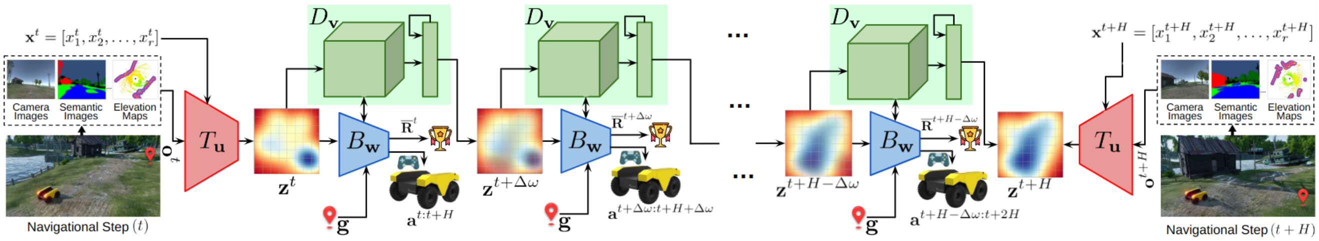


Fig. 2. Overview of our proposed approach for robot navigation in unstructured environments. The traversability encoder network $T_\mathbf{u}$, captures robot observations and states within a low-dimensional representations. The dynamics inference network $D_\mathbf{v}$ learns to predict how these representations change with navigational behaviors. Finally, through the behavior generation network $B_\mathbf{w}$, learn to predict future rewards using these representations and generate optimal navigational behaviors to reach goal positions.

reduce errors and enables parallel training, while still able to capture the complete robot-terrain interaction dynamics.

Formally, robot observations from RGB-D images, and LiDAR readings, acquired by the robot at time t, are denoted as $\mathbf{o}^t = \{\mathbf{o}_1^t, \ldots, \mathbf{o}_m^t\}$, where $\mathbf{o}_i^t \in \mathbb{R}^{k_i}$ represents the observation obtained from the i-th modality at time step t and k_i denotes its dimensionality. Additionally, we denote the robot's current state, including pose, velocity, and acceleration, acquired by Inertial Measurement Unit and wheel odometers as $\mathbf{x}^t \in \mathbb{R}^r$, where r indicates its dimensionality.

To predict the traversability representations, we aim to develop a traversability encoder network, $T_\mathbf{u} : \mathbf{o}^t, \mathbf{x}^t \to \mathbf{z}^t$, parameterized by $\mathbf{u}$, to learn stochastic traversability representations, $\mathbf{z}^t \in \mathbb{R}^q$ $(q \ll k_i)$, from a given set of terrain observations, $\mathbf{o}^t$, and robot states, $\mathbf{x}^t$. These representations, while not directly corresponding to physical quantities, encode essential information about the terrain in a significantly lower dimensional observation space. Modeling traversability using stochastic representations also enables the capture of uncertainty and variability in the terrain. We design $T_\mathbf{u}$ in a manner that the input sensory observations and robot states are processed using a series of convolutional layers, the output of which is processed by fully connected layers to output traversability representations.

We then learn to estimate how these representations change as the robot traverses the terrain. We denote robot navigational behaviors, i.e., behavior controls (e.g., linear and angular velocity), as $\mathbf{a}^t \in \mathbb{R}^p$, where p is the dimensionality of the individually controllable behaviors. Formally, we employ a dynamics inference network $D_\mathbf{v} : \mathbf{z}^t, \mathbf{a}^t \to \mathbf{z}^{t+\Delta t}$, parameterized by weights $\mathbf{v}$, to predict future traversability representations using the current representations, $\mathbf{z}^t$, and robot navigation behaviors, $\mathbf{a}^t$.

Then the problem of learning to infer robot-terrain interaction dynamics within the traversability representation space is defined as:

$$\min_{\mathbf{u},\mathbf{v}} \sum_{\tau=t}^{t+L} \mathbb{E}\left[KL\Big(T_\mathbf{u}(\mathbf{o}^\tau, \mathbf{x}^\tau) \big\| D_\mathbf{v}(\tilde{\mathbf{z}}^{\tau-\Delta\tau}, \mathbf{a}^{\tau-\Delta\tau})\Big)\right] \quad (1)$$
$$- \sum_{\tau=t}^{t+L} \mathbb{E}\Big[\mathcal{E}(T_\mathbf{u}(\mathbf{o}^\tau, \mathbf{x}^\tau)\Big]$$

where L denotes the time length of robot navigational episode used during training. Eq. (1) aims to minimize the expected Kullback-Leibler (KL) divergence [56] between the distribution of representations generated by the traversability encoder network, $T_\mathbf{u}$, and from the dynamics network, $D_\mathbf{v}$. Specifically, Eq. (1) minimizes the expected difference between probability distributions, i.e., $\hat{\mathbf{z}}^\tau \sim T_\mathbf{u}(\mathbf{o}^\tau, \mathbf{x}^\tau)$ and $\tilde{\mathbf{z}}^\tau \sim D_\mathbf{v}(\tilde{\mathbf{z}}^{\tau-\Delta\tau}, \mathbf{a}^{\tau-\Delta\tau})$.

In Eq. (1), the first term serves a dual purpose. It enables the dynamics inference network to predict the next representation using reference from the traversability encoder network. This consistency is crucial for the robot to maintain a precise understanding of the evolving terrain it interacts with during the navigation task. In addition, this term ensures that the traversability representations generated by $T_\mathbf{u}$ selectively capture essential robot and terrain observations that change with the robot's movements. This allows our approach to retain information about only navigation specific terrain observations and robot states. By minimizing this expected KL loss, we guarantee that the estimated traversability representations from the dynamics inference model closely align with future representations recorded by $T_\mathbf{u}$.

The second term in Eq. (1) represents the expected entropy value of the generated traversability representations, where $\mathcal{E}(\cdot)$ is the entropy operator. This term increases the cost when the learned representations exhibit excessive concentration and encourages the agent to generate diverse representations.

While the learned traversability representations capture knowledge about the dynamics of robot-terrain interactions, the robot still is unable to leverage traversability representations and generate navigational behaviors to reach a goal position.

B. Emulating Rehearsals for Learning Navigational Behaviors

We introduce in our approach the ability to generate navigational behaviors by emulating rehearsals based on the dynamics inference model. This rehearsal process involves the robot simulating future scenarios within the traversability representation space, and predicting its reward to learn better navigational behaviors

We formulate the problem of navigational behavior generation for unstructured terrain navigation as a γ-discounted partially observable Markov Decision Process (POMDP) aimed at planning robot navigational behaviors within the traversability representation space. For the task of navigating to a goal position, we define the cumulative reward function to represent the immediate rewards the robot earns after executing a behavior at time $t-1$ as follows:

$$R^t(\mathbf{z}^t, \mathbf{g}, \mathbf{z}^{t-\Delta t}) = \lambda_1 r^t_{goal} + \lambda_2 r^t_{align} + \lambda_3 r^t_{obs} \quad (2)$$

where r_{goal} denotes the reward obtained by the robot as it navigates to the goal position. Mathematically, $r^t_{goal} = d^{t-1} - d^t$, with d^t representing the Euclidean distance between the robot and the goal position, $\mathbf{g}$. The second component of the reward function, r_{align}, encourages the robot to maintain a consistent heading towards a goal position and is expressed as $r^t_{align} = \max(-1, -\theta^t)$, with θ^t as the heading of the robot towards $\mathbf{g}$. The third term in our reward function encourages the robot to avoid obstacles in its path, expressed as follows:

$$r^t_{obs} = \sum_{i=1}^{n} \begin{cases} -\frac{1}{d^t_i} & \text{if } d^t_i \leq d_{\min} \\ 0 & \text{if } d^t_i > d_{\min} \end{cases}$$

where d_i denotes the distance to the i-th obstacle, and n denotes the number of different obstacles at time t. In Eq. (2), the parameters λ_1, λ_2, and λ_3 represent the normalized weights that model the relative importance between different aspects of the reward function.

Given the reward function R^t, we learn a behavior generation network $B_{\mathbf{w}} = \{B_{\mathbf{w}_A}, B_{\mathbf{w}_C}\}$, that comprises an actor $B_{\mathbf{w}_A}$, and a critic model $B_{\mathbf{w}_C}$. The actor component of the network learns a distribution over successful and efficient navigational behaviors that maximize the sum of predicted future navigational task rewards. Specifically, $B_{\mathbf{w}_A}$: $\mathbf{z}^t, \mathbf{g} \rightarrow \left(\mathbf{a}^{t:t+H} \mid \arg\max_{\mathbf{a}^{t:t+H}} \sum_{\tau=t}^{t+H} R^\tau(\cdot)\right)$, where H denotes the number of future navigational behaviors we plan to execute towards reaching the goal position, i.e., the control horizon. On the other hand, the critic model $B_{\mathbf{w}_C}$: $\mathbf{z}^t, \mathbf{g} \rightarrow \bar{R}^t$ learns to estimate the average rewards $\bar{R}$ given the robot's navigational goals and present traversability representations. Mathematically, we express $\bar{R}^t \approx \mathbb{E}_{\mathbf{z} \sim D_{\mathbf{v}}(z^{t-\Delta t}, a^{t-\Delta t})} \left[R(\mathbf{z}^t, \mathbf{g}, \mathbf{z}^{t-\Delta t}) + \gamma \bar{R}^{t+1}\right]$.

The task of learning navigational behavior via rehearsals for navigation in unstructured terrains can be achieved by maximizing the following objective:

$$\begin{aligned} \max_{\mathbf{w}} \sum_{\omega=t}^{t+H} \mathbb{E}_{\mathbf{a}^\omega \sim B_{\mathbf{w}_A}} \Big[& \gamma^{\omega-t} \ln(P(\mathbf{a}^\omega)) \Big(R^\omega(\mathbf{z}^\omega, \mathbf{g}, \mathbf{z}^{\omega-\Delta\omega}) \\ & - \bar{R}^\omega \Big) + \mathcal{E}(B_{\mathbf{w}_A}(\mathbf{z}^\omega, \mathbf{g})) \Big] \\ \text{s.t.} \quad \min_{\mathbf{u},\mathbf{v}} \quad & \mathbb{E}\left[\sum_{\tau=t}^{t+L} KL\Big(T_{\mathbf{u}}(\mathbf{o}^\tau, \mathbf{x}^\tau) \big\| D_{\mathbf{v}}(\hat{\mathbf{z}}^{\tau-\Delta\tau}, \mathbf{a}^{\tau-\Delta\tau})\Big)\right] \\ & - \sum_{\tau=t}^{t+H} \mathbb{E}\Big[\mathcal{E}(T_{\mathbf{u}}(\mathbf{o}^\tau, \mathbf{x}^\tau)\Big] \end{aligned} \quad (3)$$

where $\Delta\omega \ll \Delta\tau$, i.e., the navigational behaviors and dynamics model are updated at much higher frequency than the traversability representation module. Here $P(\cdot)$ represents the probability distribution function. The loss function in Eq. (3) is composed of two terms: a policy loss term and a latent loss term. The policy loss term encourages the agent to select actions that lead to higher expected returns. Specifically, the term $\left(R^\omega(\mathbf{z}^\omega, \mathbf{g}, \mathbf{z}^{\omega-\Delta\omega}) - \bar{R}^\omega\right)$ encourages the behavior generation network to prioritize actions that yield higher rewards. This term is weighted by a factor of γ, reflecting its diminishing importance as the time horizon increases. The term is also multiplied by the natural logarithm of the action probability, not favoring less likely actions.

In Eq. (3), the entropy of the action distribution represents the agent's uncertainty about which action to take. High entropy indicates exploration of different actions, while low entropy suggests repetitive actions. By including this term in the loss function, we encourage the agent to learn a policy that explores novel actions, facilitating the exploration of better navigational behaviors in the environment.

We optimize the behavior generation network, the traversability representation network, and the dynamics inference network using Adam optimizer [57]. We follow a two-step process when the robot is navigating in an environment to learn the parameters of our approach. In the first step, we optimize the traversability representation network and dynamics inference network. In the second step, keeping other models fixed, we learn the behavior generation network.

Once optimal parameters $\mathbf{u}^*$, $\mathbf{v}^*$, and $\mathbf{w}^*$ are learned, during the execution stage, given robot observations $\mathbf{o}^{t_0}$, robot states $\mathbf{x}^{t_0}$, and goal positions $\mathbf{g}$, robot navigational behaviors at time t can be executed as:

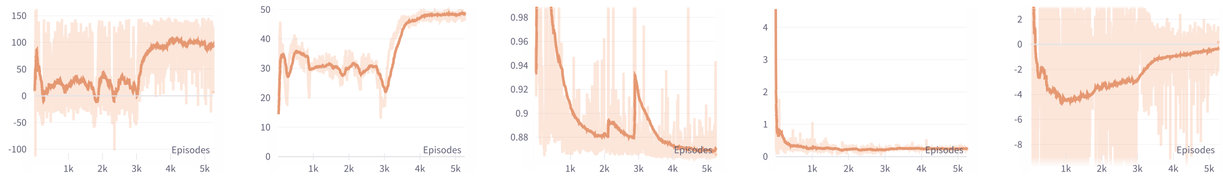
$$\mathbf{a}^{t:t+H} \sim B_{\mathbf{w}^*}(\mathbf{z}^t, \mathbf{g}); \quad \mathbf{z}^t \sim \begin{cases} T_{\mathbf{u}^*}(\mathbf{o}^{t_0}, \mathbf{x}^{t_0}) & \text{if } t = t_0 \\ D_{\mathbf{v}^*}(\hat{\mathbf{z}}^{t-1}, \mathbf{a}^{t-1}) & \text{if } t > t_0 \end{cases}$$

where $\mathbf{a}^{t:t+H}$, denotes the navigational behavior plan for the next H time window. We apply the navigational behaviors $\mathbf{a}^t$ to the robot and estimate the plan at the next interval according to the evolved trajectory representations through the dynamics model until new observations are available. This working of our approach is illustrated in Figure 2.

IV. Experiments

We train and evaluate our RIDER approach for autonomous navigation using a simulated Clearpath Warthog unmanned ground vehicle (UGV) in unstructured environments. This robot is equipped with exteroceptive sensors including an Intel Realsense D435 color depth camera and an Ouster OS1-64 LiDAR to capture environmental observations. It also has proprioceptive sensors including an inertial measurement unit (IMU) and wheel encoders to capture robot state. Inputs to our approach include RGB-D images from the camera, grid-based elevation maps obtained from LiDAR, 6DOF pose, velocity and acceleration from the wheel encoders and IMU, and 6DOF robot goal positions. These inputs are collected at 15Hz using linear interpolation.

Our training setup consists of an Intel i9 workstation with 16GB RAM, Nvidia RTX 3080 GPU with 8GB VRAM, and Ubuntu 20.04 with Robot Operating System (ROS) Noetic and PyTorch. We train our approach for 5,100 episodes totaling 200,000 observations over approximately 5 hours. For testing our approach, we use the learned model on the same workstation, which consistently generates navigational behaviors at 45Hz, even when robot observations drop to 8Hz -10 Hz.



(a) Accumulated Rewards (b) Expected Rewards (c) Reward Prediction Loss (d) KL Divergence Loss (e) Behavior Generation Loss

Fig. 3. Plots to depict the training progression of our approach.

To train our approach, we use a curriculum-based reward with dynamically adjusted weights ($\lambda_1, \lambda_2, \lambda_3$) based on training progress. Specifically, we use a three step curriculum with $\lambda_1, \lambda_2, \lambda_3$ as $\{0.1, 0.7, 0.2\}, \{0.1, 0.4, 0.5\}, \{0.4, 0.2, 0.4\}$ for the initial, middle, and final training stages. In the initial stage, goal alignment is given higher weight. In the middle stage, we focus more on enhancing obstacle avoidance. Finally, in the last stage, navigating to the goal and obstacle avoidance are given equally high importance. The learning progression of our RIDER approach is further illustrated in Figure 3 to investigate its characteristics. Figure 3(a) shows the accumulated reward for each training episode, where we see oscillating rewards initially as the approach gets stuck in local minima and due to curriculum learning. Figure 3(b) plots the expected rewards that the critic model forecasts, setting a baseline to generate better-rewarded behaviors. The critic's reward prediction loss in Figure 3(c) depicts its error in estimating average rewards. We specifically observe increases in this loss when curriculum shifts occur around episodes 2,000 and 3,000, followed by optimization for the new rewards. Since rewards do not affect traversability representations in our approach, the KL loss between the traversability encoder and dynamics inference network constantly decreases, unaffected by curriculum learning as shown in Figure 3(d). Finally, Figure 3(e) shows the complete behavior generation network's error in learning to predict navigational behaviors.

We validate our approach on long distance navigation tasks across two distinct simulated environments as shown in Figure 4. The first environment is an urban setting comprising concrete, mud, and grass terrain, as well as dense areas of houses, walls, cars, street lamps and trees as obstacles. The second environment is a highly unstructured off-road marshland with bodies of water. In addition to the challenges of navigating the urban setting, this marsh environment contains abrupt elevation changes, fences, and other obstacles that occlude the terrain. The occlusions coupled with the uneven and wet terrain make navigation particularly difficult in the marsh environment. Both environments provide diversity in terrains and obstacles to comprehensively evaluate our method's autonomous navigation capabilities.

We compare our approach to three baselines comprising classical control and learning-based navigation methods. The control method is Model Predictive Path Integral (MPPI) control [6] with two variants: a highly conservative obstacle avoidance model and a greedy minimum time goal reaching model. The learning methods are Terrain Representation and Apprenticeship Learning (TRAL) [31] and Enhancing Consistent Ground Maneuverability (ECGM) [11]. We train TRAL and ECGM in our simulation environments for a fair comparison. All baselines use the original hyperparameter settings from their papers. Futhermore, even though the robot is capable of navigating at a max speed of 5 m/s, we limit the max speed output from these methods to 3 m/s since the baseline methods were trained at this speed.

To quantitatively evaluate and compare performance of the different methods, we use the following evaluation metrics:

- *Failure rate (FR)*: this metric is defined as the number of times the robot fails to complete the navigational task across a set of experimental trials. A failure occurs when the robot gets stuck in the terrain, collides with an obstacle or flips over in the terrain. Lower values of failure indicate better performance and is preferred.
- *Traversal Time (TT)*: this metric is defined as the average time taken to complete the navigational task in all the successful runs. A smaller value for traversal time indicates better performance and may be preferred.
- *Distance traveled (DT)*: this metric is defined as the average distance traveled by the robot while reaching the goal position across all successful runs. Generally, a smaller value for the distance travel metric indicates a better performance and may be preferred.

A. Navigation in Cluttered Urban Environments

In this set of experiments, we assess the performance of our approach and the baseline methods on a long-range navigation task in a cluttered urban environment. The robot is trained within one section of the environment and evaluated on a 450m navigation task in a different, unseen section. To evaluate performance, we conduct five evaluation runs with identical conditions across approaches and the environment. We calculate the evaluation metrics by averaging over successful runs for each of the methods.

The quantitative results obtained by the different methods are shown in Table I. In this environment, robot navigational

Cluttered Urban Environment Unstructured Off-Road Environment

Fig. 4. Unstructured environments used for experimentation.

failures occur when the robot encounters small obstacles (e.g., electric light poles and trees) and cannot generate a path around them, thus getting stuck in the terrain. Accordingly, in terms of failure rate we observe that MPPI-S and TRAL exhibit the highest failure rates, indicating poor performance. On the other hand, the ECGM method demonstrates better performance by achieving fewer failures. In contrast to MPPI-S, MPPI-G experiences fewer failures. This can be attributed to the fact that MPPI-G is less sensitive to small obstacles in the terrain and can navigate very close to these obstacles in order to reach the goal position effectively. RIDER achieves a significantly lower failure rate compared to other methods, highlighting its robustness in handling small obstacles while navigating in cluttered environments.

Analyzing both distance traversed and the traversal time metric, we observe that MPPI-S, in contrast to MPPI-G, tends to opt for longer paths to avoid obstacles, resulting in a longer traversal time to reach the goal. However, it's worth noting that MPPI-S, with only one successful run, lacks sufficient data for presenting standard deviations in the evaluation metrics. The TRAL approach does the best in minimizing the distance traversed, taking shorter paths to the goal. However, this efficiency comes at the cost of a higher failure rate, as the approach frequently collides with obstacles on its path. RIDER, while not achieving the shortest distance traversed, consistently accomplishes the navigation task in less time. This efficiency stems from RIDER's ability to generate navigational behaviors by estimating both the future robot-terrain interactions and accordingly planning the most effective navigational behaviors to reach goal positions.

B. Navigating over Unstructured Off-road Environments

In this set of experiments, we assess the performance of different methods on a long-range navigation task in a highly unstructured off-road environment. No further training is performed for these experiments as we use the same model learned for the urban environment. The different methods are tasked with navigating from an initial position to a goal position approximately 300m away.

The quantitative results of the different methods are presented in Table II. In this environment, robot navigation failures may occur due to wheel slip, collisions with narrow obstacles like fences or shrubs, or getting stuck in trenches when a viable path cannot be found. When considering the failure rate metric, we observe that MPPI-S experiences fewer failures in this environment, while MPPI-G does not achieve any successful runs. In both versions of MPPI, failures occur when the robot became stuck in trenches. Similar to the previous set of experiments, TRAL completes only one successful run, and ECGM completes two successful runs, with failures resulting from collisions when navigating in close proximity to obstacles. Once again, RIDER achieves the lowest failure rate. Failures for RIDER specifically occur when navigating between narrow obstacles, such as trees.

In terms of the distance traversed and traversal time metric, we observe a similar trend to the first set of experiments. The MPPI-S approach achieves the highest traversal time metric and also had higher distance traversed metrics, as it follows a path with very conservative behaviors to avoid collisions with obstacles. Again, the TRAL approach achieves the lowest distance traversed metrics but had significantly higher failure rates. RIDER, on the other hand, achieves the highest distance traversed metric, while also having the lowest failure rates and shortest traversal time metric. Throughout the navigational task, we consistently observe that RIDER navigates at higher speeds compared to other methods and is able to reach the goal position the fastest.

TABLE I
QUANTITATIVE RESULTS FOR SCENARIOS WHEN THE ROBOT NAVIGATES IN CLUTTERED URBAN ENVIRONMENTS. SUCCESSFUL RUNS (WITH NO NAVIGATION FAILURES) ARE USED TO CALCULATE THE METRICS OF TRAVERSAL TIME AND DISTANCE TRAVERSED.

Method	FR (/5)	DT (m)	TT (s)
MPPI-S	4	581.87 (±0.00)	329.24 (±0.00)
MPPI-G	2	567.11 (±16.04)	321.73 (±11.56)
TRAL	4	**552.98 (±0.00)**	306.11 (±0.00)
ECGM	3	559.25 (±11.98)	315.07 (±7.80)
RIDER	**1**	569.23 (±11.79)	**273.21 (±7.67)**

TABLE II
QUANTITATIVE RESULTS FOR SCENARIOS WHEN THE ROBOT NAVIGATES OVER THE UNSTRUCTURED OFF-ROAD ENVIRONMENTS. SUCCESSFUL RUNS (WITH NO NAVIGATION FAILURES) ARE USED TO CALCULATE THE METRICS OF TRAVERSAL TIME AND DISTANCE TRAVERSED.

Method	FR (/5)	DT (m)	TT (s)
MPPI-S	3	393.16(±17.25)	252.03 (±11.53)
MPPI-G	5	-	-
TRAL	4	**380.07 (±0.00)**	207.11 (±0.00)
ECGM	3	387.61 (±12.68)	234.34 (±5.35)
RIDER	**2**	414.02 (±4.10)	**199.97 (±5.80)**

V. CONCLUSION

In this paper, we present our novel RIDER approach for autonomous robot navigation in unstructured environments. RIDER learns a compact traversability representation that captures critical aspects of robot-terrain interactions for navigation. By predicting future traversability representations in response to navigational behaviors, RIDER acquires the capacity to infer the dynamics of robot-terrain interactions, enabling the robot to anticipate the impact of its behaviors on future observations and states. Finally, through the process of emulated rehearsals, RIDER predicts future rewards and plans behaviors within the traversability representation space without relying on real observations. This allows for executing navigational behaviors even when sensory observations are not immediately available. Together, these contributions provide RIDER with a framework to efficiently learn intelligent navigation behaviors and execute them at higher rates, as it processes only a small observation space, which itself updates faster than real observations. Experimental results also validate that RIDER outperforms prior methods in complex unstructured environments.

REFERENCES

[1] H.-T. L. Chiang, B. HomChaudhuri, L. Smith, and L. Tapia, "Safety, Challenges, and Performance of Motion Planners in Dynamic Environments," in *Robotics Research*, 2020, pp. 793–808.

[2] A. Mazumdar, M. Y. Chuah, M. S. Triantafyllou, and H. H. Asada, "Design for Precision Multi-Directional Maneuverability: Egg-Shaped Underwater Robots for Infrastructure Inspection," in *International Conference on Robotics and Automation*, 2014.

[3] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima *et al.*, "Emergency Response to the Nuclear Accident at the Fukushima Daiichi Nuclear Power Plants using Mobile Rescue Robots," *Journal of Field Robotics*, vol. 30, no. 1, pp. 44–63, 2013.

[4] R. Siegwart, M. Hutter, P. Oettershagen, M. Burri, I. Gilitschenski, E. Galceran, and J. Nieto, "Legged and Flying Robots for Disaster Response," in *World Engineering Conference and Convention*, 2015.

[5] U. Iqbal, A. Noureldin, J. Georgy, and M. J. Korenberg, "Application of System Identification Techniques for Integrated Navigation," in *International Conference on Communications, Signal Processing, and their Applications*, 2021.

[6] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive Driving with Model Predictive Path Integral Control," in *International Conference on Robotics and Automation*, 2016.

[7] Q. Zhu, Y. Han, C. Cai, and Y. Xiao, "Robust Optimal Navigation Using Nonlinear Model Predictive Control Method Combined With Recurrent Fuzzy Neural Network," *Mathematical Problems in Engineering*, 2018.

[8] U. Nehmzow, O. Akanyeti, C. Weinrich, T. Kyriacou, and S. A. Billings, "Robot Programming by Demonstration through System Identification," in *International Conference on Intelligent Robots and Systems*, 2007.

[9] M. Wigness, J. G. Rogers, and L. E. Navarro-Serment, "Robot Navigation from Human Demonstration: Learning Control Behaviors," in *International Conference on Robotics and Automation*, 2018.

[10] Z. Wang, X. Xiao, B. Liu, G. Warnell, and P. Stone, "APPLI: Adaptive Planner Parameter Learning from Interventions," in *International Conference on Robotics and Automation*, 2021.

[11] S. Siva, M. Wigness, J. Rogers, and H. Zhang, "Enhancing Consistent Ground Maneuverability by Robot Adaptation to Complex Off-Road Terrains," in *Conference on Robot Learning*, 2021.

[12] G. Kahn, A. Villaflor, P. Abbeel, and S. Levine, "Composable Action-conditioned Predictors: Flexible Off-policy Learning for Robot Navigation," in *Conference on Robot Learning*, 2018.

[13] B. Sofman, E. Lin, J. A. Bagnell, J. Cole, N. Vandapel, and A. Stentz, "Improving Robot Navigation Through Self-supervised Online Learning," *Journal of Field Robotics*, 2006.

[14] F. Fathinezhad, V. Derhami, and M. Rezaeian, "Supervised Fuzzy Reinforcement Learning for Robot Navigation," *Applied Soft Computing*, 2016.

[15] G. Kahn, A. Villaflor, B. Ding, P. Abbeel, and S. Levine, "Self-supervised Deep Reinforcement Learning with Generalized Computation Graphs for Robot Navigation," in *International Conference on Robotics and Automation*, 2018.

[16] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "RMA: Rapid Motor Adaptation for Legged Robots," *Robotics: Science and Systems*, 2021.

[17] Z. Wu, A. Tran, D. Rincon, and P. D. Christofides, "Machine Learning-based Predictive Control of Nonlinear Processes. Part I: Theory," *AIChE Journal*, 2019.

[18] S. Yang, M. P. Wan, W. Chen, B. F. Ng, and S. Dubey, "Model Predictive Control with Adaptive Machine-Learning-Based Model for Building Energy Efficiency and Comfort Optimization," *Applied Energy*, 2020.

[19] T. Duriez, S. L. Brunton, and B. R. Noack, *Machine Learning Control-Taming Nonlinear Dynamics and Turbulence*. Springer, 2017.

[20] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

[21] M. S. Masmoudi, N. Krichen, M. Masmoudi, and N. Derbel, "Fuzzy Logic Controllers Design for Omnidirectional Mobile Robot Navigation," *Applied soft computing*, 2016.

[22] H. Seraji and A. Howard, "Behavior-Based Robot Navigation on Challenging Terrain: A Fuzzy Logic Approach," *IEEE Transactions on Robotics and Automation*, 2002.

[23] M. Nazari, J. Amiryan, and E. Nazemi, "Improvement of Robot Navigation using Fuzzy Method," in *Joint Conference of AI & Robotics and 5th RoboCup Iran Open International Symposium*, 2013.

[24] Y. Aoyama, G. Boutselis, A. Patel, and E. A. Theodorou, "Constrained Differential Dynamic Programming Revisited," in *International Conference on Robotics and Automation*, 2021.

[25] S. Rahman and S. L. Waslander, "Uncertainty-Constrained Differential Dynamic Programming in Belief Space for Vision Based Robots," *Robotics and Automation Letters*, 2021.

[26] D. Fridovich-Keil, E. Ratner, L. Peters, A. D. Dragan, and C. J. Tomlin, "Efficient Iterative Linear-Quadratic Approximations for Nonlinear Multi-Player General-Sum Differential Games," in *International Conference on Robotics and Automation*, 2020.

[27] J. Truong, S. Chernova, and D. Batra, "Bi-directional Domain Adaptation for Sim2Real Transfer of Embodied Navigation Agents," *Robotics and Automation Letters*, 2021.

[28] C. J. Ostafew, A. P. Schoellig, T. D. Barfoot, and J. Collier, "Learning-Based Nonlinear Model Predictive Control to Improve Vision-based Mobile Robot Path Tracking," *Journal of Field Robotics*, vol. 33, no. 1, pp. 133–152, 2016.

[29] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Generalizing Koopman Theory to Allow for Inputs and Control," *Journal on Applied Dynamical Systems*, vol. 17, no. 1, pp. 909–930, 2018.

[30] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition," *Journal of Nonlinear Science*, vol. 25, no. 6, pp. 1307–1346, 2015.

[31] S. Siva, M. Wigness, J. Rogers, and H. Zhang, "Robot Adaptation to Unstructured Terrains by Joint Representation and Apprenticeship Learning," in *Robotics: science and systems*, 2019.

[32] A. Shaban, X. Meng, J. Lee, B. Boots, and D. Fox, "Semantic Terrain Classification for Off-road Autonomous Driving," in *Conference on Robot Learning*, 2022.

[33] C. Sevastopoulos and S. Konstantopoulos, "A survey of traversability estimation for mobile robots," *IEEE Access*, vol. 10, pp. 96 331–96 347, 2022.

[34] D. Maturana, P.-W. Chou, M. Uenoyama, and S. Scherer, "Real-time semantic mapping for autonomous off-road navigation," in *International Conference on Field and Service Robotics*, 2018.

[35] B. Gao, S. Hu, X. Zhao, and H. Zhao, "Fine-grained off-road semantic segmentation and mapping via contrastive learning," in *International Conference on Intelligent Robots and Systems*, 2021.

[36] X. Cai, M. Everett, J. Fink, and J. P. How, "Risk-aware off-road navigation via a learned speed distribution map," in *International Conference on Intelligent Robots and Systems*, 2022.

[37] R. Lenain, B. Thuilot, A. Guillet, and B. Benet, "Accurate target tracking control for a mobile robot: A robust adaptive approach for off-road motion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[38] S. Siva and H. Zhang, "Robot perceptual adaptation to environment changes for long-term human teammate following," *The International Journal of Robotics Research (IJRR)*, vol. 41, no. 7, pp. 706–720, 2022.

[39] J.-w. Hu, B.-y. Zheng, C. Wang, C.-h. Zhao, X.-l. Hou, Q. Pan, and Z. Xu, "A survey on multi-sensor fusion based obstacle detection for intelligent ground vehicles in off-road environments," *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 5, pp. 675–692, 2020.

[40] G. Kahn, P. Abbeel, and S. Levine, "LaND: Learning to Navigate from Disengagements," *Robotics and Automation Letters*, vol. 6, no. 2, pp. 1872–1879, 2021.

[41] D. Shah, B. Eysenbach, G. Kahn, N. Rhinehart, and S. Levine, "ViNG: Learning Open-World Navigation with Visual Goals," in *International Conference on Robotics and Automation*, 2021.

[42] X. Liu, Z. Cao, Y. Yu, G. Ren, J. Yu, and M. Tan, "Robot Navigation Based on Situational Awareness," *Transactions on Cognitive and Developmental Systems*, 2021.

[43] M. V. Gasparino, A. N. Sivakumar, Y. Liu, A. E. B. Velasquez, V. A. H. Higuti, J. Rogers, H. Tran, and G. Chowdhary, "WayFAST: Navigation with Predictive Traversability in the Field," *Robotics and Automation Letters*, 2022.

[44] Y. Yang, X. Meng, W. Yu, T. Zhang, J. Tan, and B. Boots, "Learning Semantics-Aware Locomotion Skills from Human Demonstration," in *Conference on Robot Learning*, 2023.

[45] X. Meng, N. Hatch, A. Lambert, A. Li, N. Wagener, M. Schmittle, J. Lee, W. Yuan, Z. Chen, S. Deng *et al.*, "TerrainNet: Visual Modeling of Complex Terrain for High-speed, Off-road Navigation," *Robotics: Science and Systems*, 2023.

[46] P. J. Schmid, “Dynamic Mode Decomposition of Numerical and Experimental Data,” *Journal of fluid mechanics*, vol. 656, pp. 5–28, 2010.

[47] J. H. Tu, “Dynamic Mode Decomposition: Theory and Applications,” Ph.D. dissertation, Princeton University, 2013.

[48] G. Mamakoukas, M. Castano, X. Tan, and T. Murphey, “Local koopman operators for data-driven control of robotic systems,” in *Robotics: science and systems*, 2019.

[49] G. L’Erario, L. Fiorio, G. Nava, F. Bergonti, H. A. O. Mohamed, E. Benenati, S. Traversaro, and D. Pucci, “Modeling, Identification and Control of Model Jet Engines for Jet Powered Robotics,” *Robotics and Automation Letters*, 2020.

[50] H. Wang and N. Noguchi, “Real-time States Estimation of a Farm Tractor using Dynamic Mode Decomposition,” *GPS Solutions*, vol. 25, no. 1, pp. 1–12, 2021.

[51] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-driven Modeling of Complex Systems*. SIAM, 2016.

[52] C. Cáceres, J. M. Rosário, and D. Amaya, “Approach of Kinematic Control for a Non-Holonomic Wheeled Robot using Artificial Neural Networks and Genetic Algorithms,” in *International Conference and Workshop on Bio-Inspired Intelligence*, 2017.

[53] D. R. Ramírez, D. Limón, J. Gomez-Ortega, and E. F. Camacho, “Nonlinear MBPC for mobile robot navigation using genetic algorithms,” in *International Conference on Robotics and Automation*, 1999.

[54] C. Della Santina, M. Bianchi, G. Grioli, F. Angelini, M. Catalano, M. Garabini, and A. Bicchi, “Controlling Soft Robots: Balancing Feedback and Feedforward Elements,” *Robotics Automation Magazine*, 2017.

[55] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, and M. D. Killpack, “Learning nonlinear dynamic models of soft robots for model predictive control with neural networks,” in *International Conference on Soft Robotics*, 2018.

[56] J. R. Hershey and P. A. Olsen, “Approximating the Kullback Leibler Divergence between Gaussian Mixture Models,” in *International Conference on Acoustics, Speech and Signal Processing*, 2007.

[57] D. P. Kingma and J. Ba, “ADAM: A Method for Stochastic Optimization,” *arXiv preprint arXiv:1412.6980*, 2014.