VertiSelector: Automatic Curriculum Learning for Wheeled Mobility on Vertically Challenging Terrain

Tong Xu, Chenhui Pan, and Xuesu Xiao

***Abstract*— Reinforcement Learning (RL) has the potential to enable extreme off-road mobility by circumventing complex kinodynamic modeling, planning, and control by simulated end-to-end trial-and-error learning experiences. However, most RL methods are sample-inefficient when training in a large amount of manually designed simulation environments and struggle at generalizing to the real world. To address these issues, we introduce *VertiSelector* (VS), an automatic curriculum learning framework designed to enhance learning efficiency and generalization by selectively sampling training terrain. VS prioritizes vertically challenging terrain with higher Temporal Difference (TD) errors when revisited, thereby allowing robots to learn at the edge of their evolving capabilities. By dynamically adjusting the sampling focus, VS significantly boosts sample efficiency and generalization within the `VW-Chrono`[1] simulator built on the Chrono multi-physics engine. Furthermore, we provide simulation and physical results using VS on a Verti-4-Wheeler platform. These results demonstrate that VS can achieve 23.08% improvement in terms of success rate by efficiently sampling during training and robustly generalizing to the real world.**

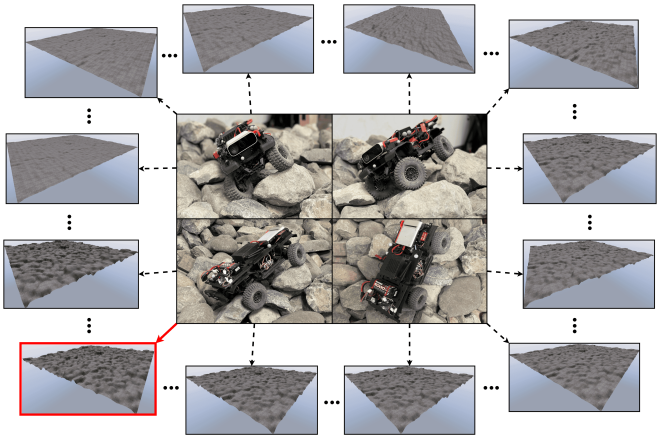

Fig. 1: VertiSelector can selectively sample training terrain based on future learning potential in the `VW-Chrono` simulator to improve RL sample efficiency and generalization.

I. Introduction

Autonomous mobile robots are increasingly being deployed in unstructured, off-road environments for applications such as search and rescue [1]–[3], planetary exploration [4]–[6], and agricultural operations [7]. However, navigating extreme terrain with dense and high vertical protrusions from the ground remains a significant challenge [8]. Traditional approaches rely on sophisticated kinodynamic modeling, motion planning, and vehicle control, which can cause cascading errors and are difficult to develop and adapt to changing conditions [9].

Reinforcement learning (RL) offers a promising alternative by enabling robots to learn end-to-end motion policies directly from simulated trial-and-error experiences [10]. By circumventing the need for explicit modeling, planning, and control, RL has the potential to achieve more robust and adaptive off-road navigation. Learning from a high-precision physics model in a simulator with RL in advance can also alleviate simulation-to-reality (sim2real) gap during deployment.

However, RL training requires a large amount of simulation data and can be sample-inefficient. It also often struggles with overfitting to the specific experiences encountered during training, which can significantly limit its ability to generalize to novel situations and hinder its broad applicability [11]. To address these limitations, Procedural Content Generation (PCG) has emerged as a promising approach [12]–[14]. PCG can algorithmically generate varied configurations before each training episode by modifying the training environments. Diverse PCG environments can improve a trained policy's generalization on previously unseen environments and can potentially form a consistent curriculum based on the RL agent's evolving capability.

To push the boundaries of off-road wheeled mobility on vertically challenging terrain, we develop a novel Automatic Curriculum Learning (ACL) method, **V**erti**S**elector (VS), which leverages differences in learning potential across various terrain produced by PCG to enhance both sample efficiency and generalization of RL. VS works in a set of PCG enviornments in a simulator, `VW-Chrono`, within the Chrono multi-physics simulation engine [15]. Throughout the training process, VS continuously assesses and updates scores that gauge the RL agent's learning potential on each terrain, taking into account its evolving capability and the Temporal Difference (TD) errors observed from the most recent trajectory sampled from that specific terrain. RL policies efficiently learned with VS in `VW-Chrono` for navigating vertically challenging terrain can then be deployed onto a physical Verti-4-Wheeler (V4W) platform [16], showing superior real-world generalizability. In summary, the contributions of this work are threefold:

- We present the `VW-Chrono` simulator (Fig. 1), designed for wheeled mobility on vertically challenging terrain to algorithmically generate varied vertically challenging terrain for ACL.

All authors are with the Department of Computer Science, George Mason University {txu25, cpan7, xiao}@gmu.edu

[1] https://github.com/RobotiXX/VWs-Chrono

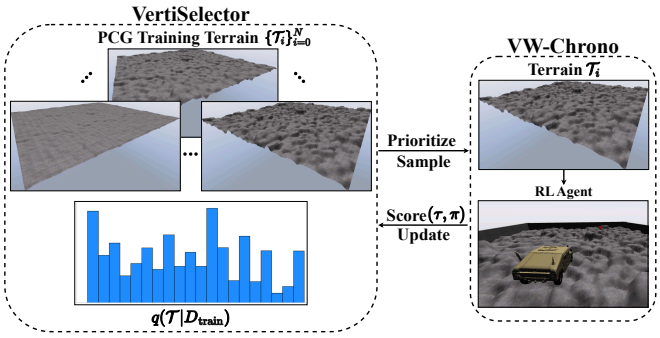

Fig. 2: VertiSelector Overview: The next training terrain $\mathcal{T}_i$ is sampled from the training distribution over the PCG-produced terrain set based on priorities determined by evaluation scores of the current policy π. A trajectory τ on this terrain $\mathcal{T}_i$ is used to update the training distribution.

- We propose VertiSelector (VS) (Fig. 2), an ACL framework that samples training terrain based on estimates of future learning potential.
- Extensive simulation and hardware experiments demonstrate that our approach significantly enhances navigation performance compared against a manually designed curriculum, vanilla RL, a hybrid (classical and learning) method and two classical baseline approaches.

II. Related Work

In this section, we review related work in off-road mobility using classical and data-driven methods, as well as curriculum learning techniques to improve learning methods.

A. Off-Road Mobility

Off-road mobility is a challenging domain for autonomous robots, as they must navigate complex, unstructured environments with varying terrain conditions. Traditional approaches to off-road navigation often rely on hand-crafted perception [17], planning [18], modeling [19], and control [20] methods with human heuristics. Those classical methods require extensive engineering effort, are affected by cascading errors from upstream modules, and struggle at adapting to new environments [21].

Considering the limitations of classical approaches, learning off-road mobility has emerged as a promising alternative avenue [22], such as learning end-to-end policies [23], semantic perception [24]–[28], kinodynamic models [29]–[34], parameter adaptation [35]–[39], and cost functions [40]–[47]. Learning methods, such as RL, can alleviate engineering effort and allow emergent and adaptive behaviors. However, those methods are often data-hungry, either requiring extensive expert demonstration or labeled data for imitation learning or millions of trial-and-error exploration steps using RL. How to generalize learning results to unseen deployment enviornments is also difficult. To tackle those challenges of learning methods, curriculum learning has the potential to improve sample efficiency and generalization by presenting the robot with a sequence of tasks that gradually increase in difficulty. VS is based on RL guided by an efficient curriculum for wheeled mobility on vertically challenging terrain.

B. Curriculum Learning

Curriculum learning is a concept inspired by the structured nature of human learning [48]. This idea was expanded upon by Bengio et al. [49], who proposed a learning paradigm where training examples are presented in a meaningful order, gradually increasing in complexity. Over the following years, curriculum learning found applications in various supervised learning settings, such as natural language processing [50] and computer vision [51]. Building on these foundational ideas, the community developed a set of mechanisms collectively known as Automatic Curriculum Learning (ACL) [52].

ACL techniques automatically adjust the distribution of training data by selecting learning situations that match the evolving capabilities of the learning agents. While ACL has been successfully applied to various domains, most applications have been limited to simple tasks or simulated environments. For instance, in supervised learning, ACL has been employed to improve performance on static benchmark datasets, such as image classification [49]. Similarly, in RL, ACL has been primarily studied in the context of simple gridworld environments [53] and Atari games [54].

Despite ACL's potential in improving sample efficiency and asymptotic performance in these simplified settings [55], few works have explored its application to real-world mobility tasks, where robots must learn to navigate complex, unstructured, off-road environments. The challenges posed by off-road terrain require sophisticated approaches to curriculum learning. In this work, we investigate how to automatically select an appropriate task sequence as a curriculum based on real-world data to enhance sample efficiency and policy generalization for wheeled robots navigating vertically challenging terrain, while considering their evolving capabilities.

III. Method

In this section, we introduce the `VW-Chrono` simulator (Sec. III-A) and its corresponding PCG environments (Sec. III-B). We also present our RL problem formulation for vertically challenging terrain in Sec. III-C and sample efficient ACL framework, VertiSelector (VS), in Sec. III-D, which considers robot's future learning potential on different terrain.

A. `VW-Chrono`

To create a realistic simulation environment for vertically challenging terrain, we first collect elevation map data [56] using our physical Verti-4-Wheeler (V4W) on a custom-built indoor testbed. This testbed consists of hundreds of randomly distributed and stacked rocks and boulders, with an average size of 30cm, matching the scale of the V4W. The test course measures 3.1×1.3m, with the highest elevation reaching up to 0.5m, more than twice the height of the vehicle (Fig. 1 middle). We generate a 150×150 synthetic

grayscale elevation map as part of state space to represent real terrain elevation distribution.

Within the Chrono multi-physics simulation engine, we generate a triangular mesh by assigning a vertex to each pixel of the elevation map. The mesh vertices are then vertically adjusted to align with the specified elevation values in the elevation map. Finally, the mesh is scaled to match the given spatial extents (Fig. 1 around). To simulate the interaction between the terrain and the vehicle's wheels, we set the friction coefficient of the terrain material using Chrono's built-in material properties. The friction coefficient is set to 0.9, and the restitution coefficient is set to 0.01. These values are carefully chosen to represent a realistic off-road terrain surface, considering factors such as tire grip and surface deformation. By following this methodology, we create a highly accurate and realistic simulation environment that closely mimics the vertically challenging terrain encountered by the V4W in real-world scenarios.

B. PCG Environments

The diversity of PCG environments makes them valuable testbeds for evaluating the robustness and generalization ability of RL agents. To maintain the PCG principle, we assign a fixed identifier (index) to each terrain. We generate a sequence of elevation maps by linearly interpolating between a starting map I_0 (flat terrain) and an ending map I_N (real-world rugged terrain) using a weighted average. The intermediate map I_i at index i out of $N+1$ indices can be calculated using the following equation:

$$I_i = (1 - \frac{i}{N})I_0 + \frac{i}{N}I_N, \quad \forall i \in \{0, 1, ..., N\}. \tag{1}$$

The $N+1$ elevation maps generated by PCG serve as individual tasks that, when ordered appropriately, comprise our curriculum to learn wheeled mobility on vertically challenging terrain.

C. POMDP for Vertically Challenging Terrain

In this work, we formulate the off-road navigation task for wheeled robots as a Partially-Observable Markov Decision Process (POMDP) characterized by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, r, \mathcal{O}, \Omega)$, where $\mathcal{S}$ represents the complete state space, $\mathcal{A}$ denotes the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathbb{P}(\mathcal{S})$ signifies the transition probability function, $\gamma \in [0, 1)$ is the discount factor, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\mathcal{O}$ is the observation space consisting of local terrain and current vehicle information, and $\Omega : \mathcal{S} \times \mathcal{A} \to \mathbb{P}(\mathcal{O})$ is the observation probability function that maps states and actions to observations. We employ RL to learn a policy $\pi : \mathcal{O} \to \mathcal{A}$ that maps observations $o \in \mathcal{O}$ to actions $a \in \mathcal{A}$, enabling the robot to navigate vertically challenging terrain while avoiding pitfalls such as getting stuck or rolling over, ultimately guiding it to reach the designated goal. The objective is to maximize the expected cumulative discounted reward:

$$J(\pi) = \mathbb{E}_{\tau \sim \pi}\left[\sum_{t=0}^{\infty} \gamma^t r(o_t, a_t)\right], \tag{2}$$

where $\tau = (o_0, a_0, o_1, a_1, \ldots)$ represents a trajectory sampled from the policy π.

To be specific, our observation space $\mathcal{O}$ includes the angular difference between the vehicle and goal heading, current vehicle velocity, and a low-dimensional representation of the elevation map patch underneath the robot obtained using a Sliced-Wasserstein Autoencoder (SWAE) [57]. The action space $\mathcal{A}$ consists of the desired linear speed and steering angle, which will be tracked by a low-level PID controller. We employ Proximal Policy Optimization (PPO) [58] as the RL algorithm to learn the policy, considering the continuous action space.

The reward function r is designed to incentivize the robot's progress toward the goal while penalizing immobilization due to excessive roll and pitch angles, as well as timeouts. It consists of three key components: (1) a progress reward that encourages the robot to move toward the goal, (2) an instability penalty that discourages excessive roll and pitch angles, and (3) a timeout penalty that penalizes the robot for not reaching the goal within a specified time limit.

By carefully designing the POMDP and reward function, we aim to learn a robust policy that enables a wheeled robot to navigate vertically challenging terrain efficiently and safely. The learned policy is then used as a foundation for our ACL approach, which further enhances RL sample efficiency and the robot's performance and generalization capabilities.

D. VertiSelector (VS)

VertiSelector (VS), depicted in Fig. 2, maintains a dynamic, non-parametric sampling distribution $q(\mathcal{T}|\mathcal{D}_{\text{train}})$ over the set of PCG-generated training terrain $\mathcal{D}_{\text{train}}$, favoring terrain with higher learning potential. Specifically, throughout the training process, VS updates $q(\mathcal{T}|\mathcal{D}_{\text{train}})$ according to a heuristic score that assigns greater weight to terrain with higher estimated learning potential based on the robot's past experiences. VS maintains two arrays, $\mathbf{u} \in \mathbb{R}^{|\mathcal{D}_{\text{train}}|}$ and $\mathbf{v} \in \mathbb{N}^{|\mathcal{D}_{\text{train}}|}$, where u_i stores the score for terrain $\mathcal{T}_i$ and v_i keeps track of the episode count at which $\mathcal{T}_i$ was last sampled. After each episode, VS updates $q(\mathcal{T}|\mathcal{D}_{\text{train}})$ by computing a mixture of two distributions: $q_u(\mathcal{T}|\mathcal{D}_{\text{train}})$, based on the terrain scores, and $q_v(\mathcal{T}|\mathcal{D}_{\text{train}})$, based on the elapsed time since each terrain was last sampled:

$$q(\mathcal{T}|\mathcal{D}_{\text{train}}) = (1 - \alpha) \cdot q_u(\mathcal{T}|\mathcal{D}_{\text{train}}) + \alpha \cdot q_v(\mathcal{T}|\mathcal{D}_{\text{train}}), \tag{3}$$

where $\alpha \in [0, 1]$ is a hyperparameter regulating the equilibrium between the two distributions. The mixture distribution ensures that the sampling process considers both the estimated learning potential and the time elapsed since each terrain was last encountered, mitigating the risk of catastrophic forgetting in neural networks.

1) Terrain Scoring Mechanism: To gauge the learning potential of a terrain $\mathcal{T}_i$, VS assigns a score u_i based on the robot's experience in the most recent episode on that terrain. The score is computed using the TD error $\delta_t = r(s_t, a_t) + \gamma V(s_{t+1}) - V(s_t)$, which quantifies the discrepancy between the expected and actual returns at each timestep. Higher-magnitude TD errors suggest a greater potential for learning

from revisiting a particular state transition. VS employs the Generalized Advantage Estimator (GAE) [59] to compute the terrain scores. The GAE at timestep t is defined as:

$$\hat{A}_t = \sum_{k=t}^{T-1} (\gamma\lambda)^{k-t}\delta_k, \tag{4}$$

where $\lambda \in [0, 1]$ is a hyperparameter that governs the bias-variance trade-off of the advantage estimates and T is the episode length. The terrain score u_i is then computed as the average absolute value of the GAE over the episode:

$$u_i = \text{score}(\tau, \pi) = \frac{1}{T}\sum_{t=0}^{T-1} |\hat{A}_t|. \tag{5}$$

The absolute value of the GAE is equal to the L1 loss between the estimated and true value functions, which is a suitable measure of the learning potential. Given the terrain scores, VS defines the score-based sampling distribution $q_u(\mathcal{T}|\mathcal{D}_{\text{train}})$ using a rank-based prioritization scheme:

$$q_u(\mathcal{T}_i|\mathcal{D}_{\text{train}}) = \frac{\text{rank}(u_i)^{-\beta}}{\sum_{j=1}^{|\mathcal{D}_{\text{train}}|}\text{rank}(u_j)^{-\beta}}, \tag{6}$$

where $\text{rank}(u_i)$ is the rank of u_i among all terrain scores in descending order, and $\beta > 0$ is a hyperparameter that allows us to tune how much $\text{rank}(u_i)$ ultimately determines the resulting distribution. This rank-based prioritization ensures that the sampling process focuses more on terrain with relatively higher learning potential while still maintaining some probability of selecting lower-scored terrain.

2) Staleness-Aware Prioritization: To prevent the terrain scores from becoming outdated and to encourage revisiting previously encountered terrain, VS incorporates a staleness-aware prioritization scheme. The staleness-based sampling distribution $q_v(\mathcal{T}_i|\mathcal{D}_{\text{train}})$ is defined as:

$$q_v(\mathcal{T}_i|\mathcal{D}_{\text{train}}) = \frac{n - v_i}{\sum_{j=1}^{|\mathcal{D}_{\text{train}}|}(n - v_j)}, \tag{7}$$

where n is the total number of episodes sampled so far, and v_i is the episode count at which terrain $\mathcal{T}_i$ was last sampled. This distribution assigns higher probability to terrain that have not been recently visited, encouraging the robot to update its knowledge of previously encountered terrain.

By combining the score-based and staleness-aware prioritization schemes, VS effectively balances the exploration of terrain with high learning potential and the exploitation of acquired knowledge, leading to more efficient and effective learning in vertically challenging environments.

IV. Implementation

In this section, we present implementation details of the `VW-Chrono` simulator and the V4W physical robot.

A. Simulation Setup

An overview of the simulation environment is shown in Fig. 3. We generate 100 distinct synthetic terrain for training based on the real-world rock testbed [16] and each terrain has a fixed identifier (index) by the PCG principle. To show the generalization of VS, the test terrain are more uneven than the training terrain, as visualized in Fig. 3. We use a mobile robot with reduced double wishbone suspensions and rack-pinion steering in `VW-Chrono`. For RL training, an episode terminates if the robot reaches the designated goal or exceeds the maximum time (20s). To learn wheeled mobility on vertically challenging terrain, the following design choices are made:

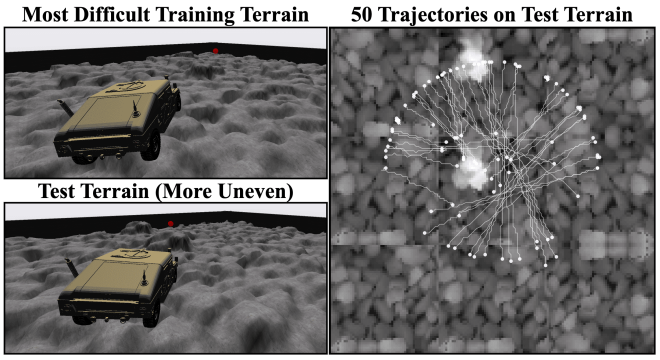

Fig. 3: Overview of the Simulation Setup with the Most Difficult Training Terrain: The test terrain is more uneven than the training ones to evaluate the generalization of VS.

1) Observation Space: The observation space consists of the components mentioned in Sec. III-C. The angular difference between the vehicle and goal heading is normalized to $[-1, 1]$, and the vehicle velocity is normalized from $[-4, 4]$ m/s to $[-1, 1]$. The 64×64 elevation map is processed by the SWAE, which reduces it to a 64×1 latent vector. This vector is further compressed to a 16×1 vector using a fully connected neural network. Finally, the compressed elevation map is concatenated with two scalar values, resulting in an 18×1 state vector.

2) Action Space: The action space, as described in Sec. III-C, includes the steering angle in the range of $[-1, 1]$ and the linear velocity normalized from $[-4, 4]$ m/s to $[-1, 1]$. A low-level PID controller generates appropriate throttle and steering commands based on these actions.

3) Policy Architecture: We employ PPO [58] with a policy network consisting of a shared feature extractor and separate fully-connected networks for the policy (actor) and value function (critic). The feature extractor takes the state space as input and passes it through a series of fully-connected layers with $\{64, 128, 64\}$ neurons and ReLU activations, outputting a 32-dimensional feature vector. Both the policy and value networks have two hidden layers with 64 neurons each and ReLU activations.

4) SWAE Architecture: The SWAE learns a compact representation of the elevation map. The encoder consists of convolutional layers with $\{32, 64, 128, 256, 512\}$ channels, kernel size 3, stride 2, padding 1, BatchNorm2d, and LeakyReLU activation, outputting a 64-dimensional latent vector mentioned above. The decoder mirrors the encoder's architecture in reversed order, with five convolutional transpose layers and channels decreasing from 512 to 32. The

final output is a reconstructed 64×64 elevation map, passed through a Tanh activation. The SWAE is trained to minimize the Sliced Wasserstein Distance between the encoded latent space and a prior distribution.

5) Reward Design: The reward function for our RL agent consists of three main components mentioned in Sec. III-C:

$$R_t = R_{\text{progress}} + R_{\text{rollover}} + R_{\text{timeout}}. \tag{8}$$

The progress reward R_{progress} incentivizes the robot to move toward the goal by providing positive rewards for the distance covered. A penalty is applied if the robot fails to move at least 1cm within 0.1 seconds:

$$R_{\text{progress}} = w_1 \cdot \Delta d - w_2 \cdot \mathbb{I}(\Delta d < 0.01), \tag{9}$$

where Δd is the distance moved toward the goal between the previous and current timestamps, $\mathbb{I}()$ is an indicator function, and w_1 and w_2 are weight terms. To prevent rollovers, the rollover penalty R_{rollover} discourages excessive roll and pitch angles:

$$R_{\text{rollover}} = -w_3 \cdot \sum_{i \in \{\text{roll,pitch}\}} \max(0, |\theta_i| - \alpha), \tag{10}$$

where θ_{roll} and θ_{pitch} are the roll and pitch angles, respectively, w_3 is a weight term and α is a constant threshold angle. Finally, the timeout penalty R_{timeout} is applied when the robot fails to reach the goal within a time limit T. It consists of a fixed penalty c and an additional penalty based on the remaining distance to the goal:

$$R_{\text{timeout}} = -(w_4 \cdot d_{\text{remain}} + c) \cdot \mathbb{I}(t \geq T), \tag{11}$$

where d_{remain} is the remaining distance to the goal, t is the current time, and w_4 is a weight term. Table I shows all hyper-parameters of our reward function.

TABLE I: Reward Weights

w_1	w_2	w_3	w_4	α	c	T
50	10	20	10	30	100	20

B. V4W and Vertically Challenging Testbed

To evaluate the performance of the learned policy, we deploy our model on the V4W platform ($0.863\text{m} \times 0.249\text{m} \times 0.2\text{m}$, Fig. 1 middle), a four-wheeled vehicle based on an off-the-shelf, two-axle, four-wheel-drive, off-road platform from Traxxas. The onboard computation is handled by an NVIDIA Jetson Xavier NX module. A Microsoft Azure Kinect RGB-D camera produces depth images to construct real-time elevation maps [56]. We use low-gear and lock both front and rear differentials to improve mobility on vertically challenging terrain. For the controlled environment, we shuffle our indoor rock testbed to achieve varying levels of difficulty: easy, medium, and hard. The testbed is designed to mimic vertically challenging terrain encountered in outdoor off-road environments with controllable complexity.

V. Experiments

We present the simulation results in the `VW-Chrono` simulator and compare the performance of VS against other baselines designed for vertically challenging terrain.

A. Baselines

VS is compared against four baseline methods: Optimistic Planner (OP), Naive Planner (NP), Vanilla RL (VR), Manually-designed Curriculum (MC) [60] and WMVCT [61].

OP minimizes the angular difference between the vehicle's current and desired heading, optimistically assuming a flat terrain. However, this assumption often struggles with steep slopes and rugged boulders, leading to suboptimal performance on vertically challenging terrain.

To address the limitations of OP, NP incorporates a heuristic based on the elevation map of the surrounding terrain. This planner divides the 64×64 surrounding elevation map into regions and selects the most traversable direction based on the mean and variance of the elevation values. Although more effective than OP, NP still relies on fixed rules and may not adapt well to diverse terrain conditions.

VR takes a more flexible approach by training the policy on randomly selected terrain from the PCG-produced training set, without any explicit curriculum. While this allows the robot to experience a wide range of terrain conditions, the lack of structure in the training process may lead to suboptimal sample efficiency, as the robot may spend too much time on terrain that is either too simple or too challenging for its current skill level.

MC addresses this issue by following a curriculum that gradually increases the difficulty of the training terrain. The curriculum consists of five stages, each with a specific success rate threshold $\{1, 1, 0.8, 0.6\}$ that the robot must reach before progressing to the next one. However, MC may not always align with the robot's actual learning progress, potentially leading to inefficiencies in the training process.

WMVCT is a hybrid method based on a sampling-based motion planner and a decomposed 6-DoF vehicle-terrain dynamics model (bicycle model for x, y, and yaw, elevation map for z, and neural network prediction for roll and pitch). Despite its high efficiency, the decomposition also introduces inaccuracies compared against a full 6-DoF model.

B. Simulation Results

Our main findings are that (i) VS with rank prioritization ($\alpha = 0.1, \beta = 0.1$) significantly improves both sample efficiency during training and generalization on the test terrain, attaining the highest success rate in 50 trials out of all baselines evaluated; (ii) The relatively low average roll and pitch angles indicate that VS learns a stable policy that can effectively handle the uneven test terrain.

1) Training Performance: As evident from the steeper slope of VS's training curve averaged over three runs in Fig. 4, it consistently achieves higher evaluation success rates with fewer training samples compared to the baselines.

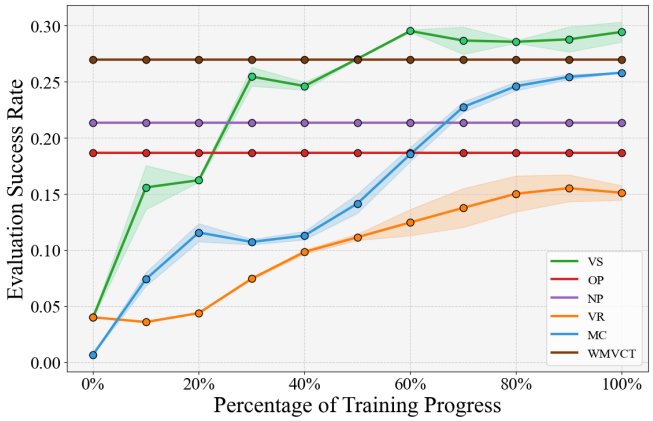

Fig. 4: Smoothed Training Curves from Test Terrain Evaluation: VS is the most sample-efficient and generalizable.

TABLE II: Simulation Experiment Results

Method	Success	Time	Angles (Roll/Pitch)
VS (Proposed)	**32/50**	8.82±4.27	**6.46±26.21** / 3.06±3.13
MC	26/50	8.46±5.19	6.55±17.56 / **2.8±1.08**
VR	23/50	8.74±10.25	8.33±45.78 / 4.47±14.46
WMVCT	20/50	7.79±0.32	8.09±22.05 / 3.75±14.27
OP	14/50	**7.56±0.37**	10.09±50.1 / 4.59±13.13
NP	13/50	8.65±0.84	8.68±28.62 / 4.81±13.43

This indicates that VS is more effective at extracting relevant information from the training terrain and updating its policy accordingly. The improved sample efficiency can be attributed to the automatic curriculum, which intelligently selects training terrain that is appropriately challenging for the robot's current skill level.

The training curve also demonstrates its superior generalization ability by converging to the highest success rate among all methods. By dynamically adjusting the difficulty, VS systematically exposes the robot to a diverse range of terrain features. This diversity helps the robot learn a more comprehensive and flexible policy that can better handle the variability and uncertainty of unseen environments. In contrast, the fixed heuristic of the planners (OP and NP), the decomposed 6-DoF vehicle-terrain dynamics model of WMVCT, the lack of a curriculum of VR, and the predefined curriculum of MC may overly specialize the robot to specific terrain types, limiting their generalization to novel test terrain.

2) Evaluation Metrics: The learned policies are evaluated on the test terrain using three metrics:

1) Number of successful trials (out of 50).
2) Mean traversal time (of successful trials in seconds).
3) Average roll/pitch angles with variance (in degrees).

Table II summarizes the performance of the best model of each method on the test terrain and the best result for each metric is shown in bold. VS achieves the highest success rate, successfully navigating the test terrain in 32 out of 50 trials. This significantly outperforms all other baselines. In terms of traversal time, VS (8.82s) is comparable to OP (7.56s). However, OP's slightly faster time comes at the cost of a drastically lower success rate (14/50). VS strikes a balance between reliable navigation and reasonable speed. Moreover, VS maintains the lowest average roll (6.46°) and second lowest pitch (3.06°) angles within the 30° threshold, demonstrating its ability to keep the vehicle stable while traversing uneven terrain.

C. Physical Results

We conduct ten trials each on three configurations of our physical testbed (Fig. 5 left) with the V4W, recording the same set of metrics. The results are presented in Table III. The learned VS policy demonstrates a high success rate across all difficulty levels, with a slight decrease in performance as the terrain complexity increases. The average traversal time also shows a consistent trend, with longer times required for more challenging courses. MC fails more on the Medium course and fails all trials on the Hard one. It mostly suffers from longer traversal time and larger roll/pitch angles. These results validate VS's generalizability from simulation to a real-world vertically challenging testbed.

TABLE III: Physical Testbed Experiment Results

Method	Difficulty	Success	Time	Angles (Roll/Pitch)
VS (Proposed)	Easy	8/10	**13.99**	**0.15/0.53**
	Medium	**7/10**	**15.85**	**2.05**/1.85
	Hard	**5/10**	**20.86**	**0.25**/8.03
MC	Easy	**9/10**	18.07	3.63/0.92
	Medium	6/10	17.22	3.97/**1.49**
	Hard	0/10	N/A	6.32/**0.9**

D. Outdoor Demonstration

To further demonstrate the generalizability and applicability of the learned VS policy, we deploy it on the V4W in a real-world outdoor environment. We select a challenging off-road location with diverse terrain features, including steep slopes, various rocks, and uneven surfaces (Fig. 5 right). The platform exhibits stable and efficient navigation by effectively making appropriate steering and throttle decisions based on the perceived outdoor terrain features.

Fig. 5: Indoor Experiments and Outdoor Demonstration.

VI. Conclusions and Limitations

This work introduces VS, an automatic curriculum learning framework that enhances sample efficiency and generalization of reinforcement learning for wheeled robot navigation on vertically challenging terrain. VS selectively samples training terrain based on the robot's evolving capabilities and learning potential to accelerate learning and facilitate robust navigation. The `VW-Chrono` simulator enables the generation of diverse and challenging terrain for training and testing. Simulation experiments demonstrate VS's superior performance compared to baseline methods, achieving a 23.08% improvement in success rate. The real-world applicability of VS is validated through successful deployment on the physical V4W platform in both an indoor testbed and outdoor environment.

One of the limitations of VS is its focus on a specific set of terrain geometry and vehicle configurations, which may limit its generalizability to more diverse off-road scenarios. Despite the effectiveness of `VW-Chrono`, sim2real gap still remains: the framework trains and evaluates using a simulated full-scale vehicle, while real-world validation employs a scaled RC car. This discrepancy in vehicle scale and dynamics may undermine the transferability of results to practical scenarios. And it is also important to note that real rocky terrain exhibits spatially varying friction and restitution properties. Future work can focus on extending VS to incorporate various terrain semantics and vehicle types, as well as integrating with other learning paradigms such as imitation learning.

References

[1] X. Xiao, E. Cappo, W. Zhen, J. Dai, K. Sun, C. Gong, M. J. Travers, and H. Choset, "Locomotive reduction for snake robots," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3735–3740.

[2] X. Xiao, J. Dufek, T. Woodbury, and R. Murphy, "Uav assisted usv visual navigation for marine mass casualty incident response," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6105–6110.

[3] R. Murphy, J. Dufek, T. Sarmiento, G. Wilde, X. Xiao, J. Braun, L. Mullen, R. Smith, S. Allred, J. Adams *et al.*, "Two case studies and gaps analysis of flood assessment for emergency management with small unmanned aerial systems," in *2016 IEEE international symposium on safety, security, and rescue robotics (SSRR)*. IEEE, 2016, pp. 54–61.

[4] K. Tiwari, X. Xiao, A. Malik, and N. Y. Chong, "A unified framework for operational range estimation of mobile robots operating on a single discharge to avoid complete immobilization," *Mechatronics*, vol. 57, pp. 173–187, 2019.

[5] K. Tiwari, X. Xiao, and N. Y. Chong, "Estimating achievable range of ground robots operating on single battery discharge for operational efficacy amelioration," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3991–3998.

[6] A. Seeni, B. Schäfer, and G. Hirzinger, "Robot mobility systems for planetary surface exploration–state-of-the-art and future outlook: a literature survey," *Aerospace Technologies Advancements*, vol. 492, pp. 189–208, 2010.

[7] A. Kumar, R. S. Deepak, D. Kusuma, and D. Sreekanth, "Review on multipurpose agriculture robot," *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, no. V, 2020.

[8] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, "Navigation planning for legged robots in challenging terrain," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1184–1189.

[9] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch *et al.*, "Anymal-a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 38–44.

[10] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[11] K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman, "Quantifying generalization in reinforcement learning," in *International conference on machine learning*. PMLR, 2019, pp. 1282–1289.

[12] K. Cobbe, C. Hesse, J. Hilton, and J. Schulman, "Leveraging procedural generation to benchmark reinforcement learning," in *International conference on machine learning*. PMLR, 2020, pp. 2048–2056.

[13] S. Risi and J. Togelius, "Increasing generality in machine learning through procedural content generation," *Nature Machine Intelligence*, vol. 2, no. 8, pp. 428–436, 2020.

[14] M. Jiang, E. Grefenstette, and T. Rocktäschel, "Prioritized level replay," in *International Conference on Machine Learning*. PMLR, 2021, pp. 4940–4950.

[15] A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischmann, M. Taylor, H. Sugiyama, and D. Negrut, "Chrono: An open source multi-physics dynamics engine," in *High Performance Computing in Science and Engineering: Second International Conference, HPCSE 2015, Soláň, Czech Republic, May 25-28, 2015, Revised Selected Papers 2*. Springer, 2016, pp. 19–49.

[16] A. Datar, C. Pan, M. Nazeri, and X. Xiao, "Toward wheeled mobility on vertically challenging terrain: Platforms, datasets, and algorithms," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 322–16 329.

[17] D. V. Lu, D. Hershberger, and W. D. Smart, "Layered costmaps for context-sensitive navigation," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 709–715.

[18] H. Rastgoftar, B. Zhang, and E. M. Atkins, "A data-driven approach for autonomous motion planning and control in off-road driving scenarios," in *2018 Annual american control conference (ACC)*. IEEE, 2018, pp. 5876–5883.

[19] R. He, C. Sandu, A. K. Khan, A. G. Guthrie, P. S. Els, and H. A. Hamersma, "Review of terramechanics models and their applicability to real-time applications," *Journal of Terramechanics*, vol. 81, pp. 3–22, 2019.

[20] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016.

[21] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[22] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion planning and control for mobile robot navigation using machine learning: a survey," *Autonomous Robots*, vol. 46, no. 5, pp. 569–597, 2022.

[23] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. A. Theodorou, and B. Boots, "Imitation learning for agile autonomous driving," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 286–302, 2020.

[24] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous robots*, vol. 18, pp. 81–102, 2005.

[25] D. Maturana, P.-W. Chou, M. Uenoyama, and S. Scherer, "Real-time semantic mapping for autonomous off-road navigation," in *Field and Service Robotics*. Springer, 2018, pp. 335–350.

[26] A. Shaban, X. Meng, J. Lee, B. Boots, and D. Fox, "Semantic terrain classification for off-road autonomous driving," in *Conference on Robot Learning*. PMLR, 2022, pp. 619–629.

[27] 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," *arXiv preprint arXiv:2303.15771*, 2023.

[28] K. Viswanath, K. Singh, P. Jiang, P. Sujit, and S. Saripalli, "Offseg: A semantic segmentation framework for off-road driving," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 354–359.

[29] X. Xiao, J. Biswas, and P. Stone, "Learning inverse kinodynamics for accurate high-speed off-road navigation on unstructured terrain," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 6054–6060, 2021.

[30] H. Karnan, K. S. Sikand, P. Atreya, S. Rabiee, X. Xiao, G. Warnell, P. Stone, and J. Biswas, "Vi-ikd: High-speed accurate off-road navigation using learned visual-inertial inverse kinodynamics," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 3294–3301.

[31] P. Atreya, H. Karnan, K. S. Sikand, X. Xiao, S. Rabiee, and J. Biswas, "High-speed accurate robot control using learned forward kinodynamics and non-linear least squares optimization," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 789–11 795.

[32] A. Datar, C. Pan, M. Nazeri, A. Pokhrel, and X. Xiao, "Terrain-attentive learning for efficient 6-dof kinodynamic modeling on vertically challenging terrain," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024.

[33] A. Pokhrel, A. Datar, M. Nazeri, and X. Xiao, "CAHSOR: Competence-aware high-speed off-road ground navigation in SE (3)," *IEEE Robotics and Automation Letters*, 2024.

[34] P. Maheshwari, W. Wang, S. Triest, M. Sivaprakasam, S. Aich, J. G. Rogers III, J. M. Gregory, and S. Scherer, "Piaug–physics informed augmentation for learning vehicle dynamics for off-road navigation," *arXiv preprint arXiv:2311.00815*, 2023.

[35] X. Xiao, B. Liu, G. Warnell, J. Fink, and P. Stone, "Appld: Adaptive planner parameter learning from demonstration," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4541–4547, 2020.

[36] Z. Wang, X. Xiao, B. Liu, G. Warnell, and P. Stone, "Appli: Adaptive planner parameter learning from interventions," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 6079–6085.

[37] Z. Wang, X. Xiao, G. Warnell, and P. Stone, "Apple: Adaptive planner parameter learning from evaluative feedback," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7744–7749, 2021.

[38] Z. Xu, G. Dhamankar, A. Nair, X. Xiao, G. Warnell, B. Liu, Z. Wang, and P. Stone, "Applr: Adaptive planner parameter learning from reinforcement," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 6086–6092.

[39] X. Xiao, Z. Wang, Z. Xu, B. Liu, G. Warnell, G. Dhamankar, A. Nair, and P. Stone, "Appl: Adaptive planner parameter learning," *Robotics and Autonomous Systems*, vol. 154, p. 104132, 2022.

[40] M. Sivaprakasam, S. Triest, W. Wang, P. Yin, and S. Scherer, "Improving off-road planning techniques with learned costs from physical interactions," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4844–4850.

[41] N. Dashora, D. Shin, D. Shah, H. Leopold, D. Fan, A. Agha-Mohammadi, N. Rhinehart, and S. Levine, "Hybrid imitative planning with geometric and predictive costs in off-road environments," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4452–4458.

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

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

[44] X. Cai, S. Ancha, L. Sharma, P. R. Osteen, B. Bucher, S. Phillips, J. Wang, M. Everett, N. Roy, and J. P. How, "EVORA: Deep evidential traversability learning for risk-aware off-road autonomy," *IEEE Transactions on Robotics*, 2024.

[45] J. Seo, S. Sim, and I. Shim, "Learning off-road terrain traversability with self-supervisions only," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4617–4624, 2023.

[46] S. Jung, J. Lee, X. Meng, B. Boots, and A. Lambert, "V-STRONG: Visual self-supervised traversability learning for off-road navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 1766–1773.

[47] X. Xiao, T. Zhang, K. M. Choromanski, T.-W. E. Lee, A. Francis, J. Varley, S. Tu, S. Singh, P. Xu, F. Xia, S. M. Persson, L. Takayama, R. Frostig, J. Tan, C. Parada, and V. Sindhwani, "Learning model predictive controllers with real-time attention for real-world navigation," in *Conference on robot learning*. PMLR, 2022.

[48] J. L. Elman, "Learning and development in neural networks: The importance of starting small," *Cognition*, vol. 48, no. 1, pp. 71–99, 1993.

[49] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[50] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[51] A. Pentina, V. Sharmanska, and C. H. Lampert, "Curriculum learning of multiple tasks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5492–5500.

[52] R. Portelas, C. Colas, K. Hofmann, and P.-Y. Oudeyer, "Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments," in *Conference on Robot Learning*. PMLR, 2020, pp. 835–853.

[53] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, "Curriculum learning for reinforcement learning domains: A framework and survey," *Journal of Machine Learning Research*, vol. 21, no. 181, pp. 1–50, 2020.

[54] N. Justesen, R. R. Torrado, P. Bontrager, A. Khalifa, J. Togelius, and S. Risi, "Illuminating generalization in deep reinforcement learning through procedural level generation," *arXiv preprint arXiv:1806.10729*, 2018.

[55] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," *Advances in neural information processing systems*, vol. 30, 2017.

[56] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter, "Elevation mapping for locomotion and navigation using gpu," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2273–2280.

[57] S. Kolouri, P. E. Pope, C. E. Martin, and G. K. Rohde, "Sliced-wasserstein autoencoder: An embarrassingly simple generative model," *arXiv preprint arXiv:1804.01947*, 2018.

[58] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[59] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[60] T. Xu, C. Pan, and X. Xiao, "Reinforcement learning for wheeled mobility on vertically challenging terrain," in *2024 IEEE International Symposium on Safety Security Rescue Robotics (SSRR)*. IEEE, 2024, pp. 125–130.

[61] A. Datar, C. Pan, and X. Xiao, "Learning to model and plan for wheeled mobility on vertically challenging terrain," *IEEE Robotics and Automation Letters*, 2024.