

VertiAdaptor: Online Kinodynamics Adaptation for Vertically Challenging Terrain

Tong Xu, Chenhui Pan, Aniket Datar, and Xuesu Xiao

Abstract—Autonomous driving in off-road environments presents significant challenges due to the dynamic and unpredictable nature of unstructured terrain. Traditional kinodynamic models often struggle to generalize across diverse geometric and semantic terrain types, underscoring the need for real-time adaptation to ensure safe and reliable navigation. We propose *VertiAdaptor* (VA), a novel online adaptation framework that efficiently integrates elevation with semantic embeddings to enable terrain-aware kinodynamic modeling and planning via function encoders. VA learns a kinodynamic space spanned by a set of neural ordinary differential equation basis functions, capturing complex vehicle-terrain interactions across varied environments. After offline training, the proposed approach can rapidly adapt to new, unseen environments by identifying kinodynamics in the learned space through a computationally efficient least-squares calculation. We evaluate VA within the Verti-Bench simulator, built on the Chrono multi-physics engine, and validate its performance both in simulation and on a physical Verti-4-Wheeler platform. Our results demonstrate that VA improves prediction accuracy by up to 23.9% and achieves a 5X faster adaptation time, advancing the robustness and reliability of autonomous robots in complex and evolving off-road environments.

I. INTRODUCTION

Off-road autonomous navigation represents one of the most challenging frontiers in robotics, where vehicles must traverse complex, unstructured environments characterized by varying terrain properties, obstacles, and dynamic conditions. State-of-the-art off-road perception and planning systems often degrade under distribution shift inevitable in real-world environments [1], [2]. As a result, reliable off-road navigation requires both enhanced terrain understanding and online kinodynamics adaptation to accommodate changing vehicle-terrain interactions [3].

To improve terrain awareness, many systems augment geometric maps with semantics information. Elevation mapping has emerged as an efficient geometric representation for ground robots, with mature real-time implementations available [4]. In parallel, new off-road datasets and platforms enable robust terrain semantic segmentation that extends beyond urban domains [5]. Recent advances in learning-based traversability estimation [6]–[8] and vision models that couple geometry with semantics (e.g., bird’s-eye view traversability [9] and multi-head terrain prediction [10]) further strengthen the integration of elevation and semantics in off-road settings. Nevertheless, unifying elevation and semantics with real-time adaptation within a single framework remains an open challenge, due to the large space and



Fig. 1: Facing a variety of constantly changing elevation and semantics on vertically challenging, off-road terrain, autonomous mobile robots need to quickly adapt their terrain understanding through online kinodynamics adaptation to achieve safe and efficient navigation.

variability of the elevation and semantic input and limited onboard computation.

To address this limitation, meta-learning offers a promising path toward rapid online adaptation. Meta-trained kinodynamic models can be quickly updated with a small amount of recent data to track changes such as new terrain or partial system failures [11]. Recent advances in continuous-time neural modeling, particularly neural Ordinary Differential Equation (ODE), provide flexible function classes for system dynamics while retaining ODE structure for integration and control [12]. When combined with neural ODE, function encoders have emerged as a powerful technique to learn compact representations of complex function spaces [13]. These methods enable efficient adaptation to new scenarios by parameterizing models within a learned basis function space, allowing for rapid online adaptation without extensive recomputation.

To push the boundaries of off-road wheeled mobility on diverse and evolving vertically challenging terrain (Fig. 1), we present *VertiAdaptor* (VA), a novel online adaptation framework that efficiently integrates elevation with semantic embeddings for terrain-aware kinodynamic modeling. Our contributions can be summarized as follows:

- A unified terrain representation using the Sliced Wasserstein Autoencoder (SWAE) [14] to efficiently obtain elevation and semantic embeddings beneath the vehicle;
- VA, the first $\mathbb{SE}(3)$ kinodynamics online adaptation framework based on complex off-road elevation and semantics that enables rapid kinodynamic model updates

Freedom (DoFs) and needs to extend the state space to $X \subset \mathbb{SE}(3)$. We define the robot state at time t as

$$\mathbf{x}_t = [x_t, y_t, z_t, \phi_t, \varphi_t, \psi_t] \in \mathbb{SE}(3),$$

where x_t, y_t, z_t represent the vehicle position in 3D, and $\phi_t, \varphi_t, \psi_t$ denote roll, pitch, and yaw. Control inputs $\mathbf{u}_t \in \mathcal{U}$ correspond to vehicle steering and speed.

The vehicle's kinodynamic behavior can be modeled in continuous time as an ODE:

$$\dot{\mathbf{x}}_t = f_\theta(\mathbf{x}_t, \mathbf{u}_t, \mathbf{e}_t),$$

where $\mathbf{e}_t \in \mathcal{E}$ denotes the environment embedding that combines terrain elevation and semantic information beneath the vehicle. For compatibility with the Model Predictive Path Integral (MPPI) [33] sampling-based control framework, we adopt a discretized form:

$$\mathbf{x}_{t+1} - \mathbf{x}_t = \int_t^{t+\Delta t} f_\theta(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{e}(\tau)) d\tau,$$

with fixed time step Δt . The function f_θ is parameterized by a set of neural ODEs [12]. The goal is to approximate forward kinodynamic model f_θ that captures nonlinear vehicle-terrain interactions while supporting efficient online adaptation in unseen terrain.

B. Elevation & Semantic Embeddings

Unlike on-road driving, the underlying elevation and semantics play a vital role in determining off-road kinodynamics on vertically challenging terrain. For example, a large boulder may cause rollover, while deformable sand may get the vehicle stuck. Therefore, our environment embedding \mathbf{e}_t unifies geometric and semantic information

$$\mathbf{e}_t = [\mathbf{e}_{\text{elev}}, \mathbf{e}_{\text{sem}}] \in \mathbb{R}^{d_e + d_s},$$

where $\mathbf{e}_{\text{elev}} \in \mathbb{R}^{d_e}$ represents local elevation features beneath the vehicle, and $\mathbf{e}_{\text{sem}} \in \mathbb{R}^{d_s}$ encodes semantic terrain properties such as friction and deformability. In our implementation, elevation is represented as 2.5D elevation maps, while semantic information is provided through bird's-eye view RGB images, both aligned with the vehicle's heading. To facilitate efficient modeling and adaptation, we use SWAE [14] to project raw elevation and semantic maps into a compact latent space.

C. Function Encoder for Neural ODEs

To create accurate kinodynamic models across diverse terrain, we represent the forward kinodynamics as a linear combination of k neural ODE basis functions:

$$\begin{aligned} \mathbf{x}_{t+1} - \mathbf{x}_t &= \sum_{i=1}^k \alpha_i \int_t^{t+\Delta t} g_i(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{e}(\tau)) d\tau \\ &= \sum_{i=1}^k \alpha_i G_i(\mathbf{x}_t, \mathbf{u}_t, \mathbf{e}_t; \theta_i), \end{aligned}$$

where $G_i(\cdot; \theta_i)$ are a set of learnable basis functions parameterized by neural ODEs and θ_i , and $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_k]^T \in \mathbb{R}^k$ are the coefficients that define the contribution of each

basis function. g_i is the basis function state change rate to be integrated into G_i . Each G_i outputs the predicted state change. In practice, we approximate the integral G_i using the fourth-order Runge-Kutta (RK4) [34] numerical integrator.

When the vehicle encounters an unseen terrain, we collect a small buffer of recent trajectories:

$$\mathcal{D}_{\text{new}} = \{(\mathbf{x}_l, \mathbf{u}_l, \mathbf{e}_l, \mathbf{x}_{l+1})\}_{l=1}^M,$$

where M is the limited number of online samples. Given the pre-trained basis functions $\{G_i\}_{i=1}^k$, the online adaptation task becomes a least squares problem to find optimal coefficients α^* . The optimal coefficients α^* can be computed efficiently in closed form as the solution:

$$\alpha^* = \begin{bmatrix} \langle G_1, G_1 \rangle & \cdots & \langle G_1, G_k \rangle \\ \vdots & \ddots & \vdots \\ \langle G_k, G_1 \rangle & \cdots & \langle G_k, G_k \rangle \end{bmatrix}^{-1} \begin{bmatrix} \langle y, G_1 \rangle \\ \vdots \\ \langle y, G_k \rangle \end{bmatrix}, \quad (1)$$

where G_i denotes the i -th basis function evaluated on the new data, y is the target state change, and $\langle \cdot, \cdot \rangle$ represents the vector inner products. This least-squares solution enables rapid online adaptation with a maximum time complexity of $O(k^3)$, far more efficient than gradient-based retraining.

D. VertiAdaptor (VA)

The VA framework consists of two components: (i) *offline training*, where neural ODE basis functions are learned across diverse environments, and (ii) *online adaptation*, where coefficients are updated repeatedly to adapt to new unseen terrain.

1) *Offline Training*: We employ a multi-step rollout approach to learn basis functions. First, with fixed basis functions, coefficients are computed by an example set. Second, the computed coefficients are used on a query set to output state changes, which are compared against ground truth to compute loss and update basis functions.

Algorithm 1 details this procedure. In each training iteration, we sample F environments and N complete trajectories per environment from the training dataset \mathcal{D} (line 6). Each environment's data are split into (the first N_{ex}) example trajectories for coefficient computation and (the remaining N_q) query trajectories to learn basis functions (lines 8-9). For each environment, coefficients $\alpha^{*(f)}$ are computed using ground truth example trajectories via Eqn. (1) (line 10). S multi-step rollouts of length T_{pred} then occur for each query trajectory n in the nested loops (lines 11-23), where each neural ODE basis function G_i output is computed via RK4 integration (line 17) and linearly combined using environment-specific coefficients (line 19). The training loss accumulates prediction errors across all rollout steps t , rollouts s , and query trajectories n (line 20) and becomes

$$\mathcal{L} = \sum_{n=1}^{N_q} \sum_{s=1}^S \sum_{t=1}^{T_{\text{pred}}} \|\mathbf{x}_{t,s,n}^{\text{pred}} - \mathbf{x}_{t,s,n}^{\text{gt}}\|_2^2,$$

where $\mathbf{x}_{t,s,n}^{\text{pred}}$ and $\mathbf{x}_{t,s,n}^{\text{gt}}$ denote the predicted and ground truth state at time t on rollout s in trajectory n respectively. This

Algorithm 1 Multi-Step Training for VA

```

1: Input: Training dataset  $\mathcal{D}$ , number of basis functions  $k$ ,
   learning rate  $\alpha$ , prediction horizon  $T_{\text{pred}}$ , rollout step size
    $\Delta t$ 
2: Output: Neural ODE basis functions  $G_1, G_2, \dots, G_k$ 
3: Initialize each basis function state change rate  $g_i$  as a
   neural network with parameters  $\theta_i$ , for  $i = 1, \dots, k$ 
4: while not converged do
5:    $L = 0$ 
6:   Sample  $F$  environments and  $N$  trajectories per envi-
     ronment from  $\mathcal{D}$ 
7:   for each environment  $f \in \{1, \dots, F\}$  do
8:      $\mathcal{T}_{\text{ex}}^{(f)} \leftarrow$  first  $N_{\text{ex}}$  trajectories
9:      $\mathcal{T}_{\text{query}}^{(f)} \leftarrow$  remaining  $N_{\text{q}}$  trajectories
10:    Compute coefficients  $\alpha^{*(f)} = [\alpha_1^{*(f)}, \dots, \alpha_k^{*(f)}]$ 
      using  $\mathcal{T}_{\text{ex}}^{(f)}$  via Eqn. (1)
11:    for each query trajectory  $n \in \mathcal{T}_{\text{query}}^{(f)}$  do
12:      for each sampled rollout  $s \in \{1, \dots, S\}$  do
13:        Initialize  $\mathbf{x}_0^{\text{pred}} \leftarrow$  initial state of rollout  $s$ 
14:        Let  $[\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{T_{\text{pred}}-1}]$  be the control se-
          quence for rollout  $s$ 
15:        for  $t = 1, \dots, T_{\text{pred}}$  do
16:          for  $i = 1, \dots, k$  do
17:             $G_i(\cdot) \leftarrow \text{RK4}(g_i, \mathbf{x}_{t-1}^{\text{pred}}, \mathbf{u}_{t-1}, \Delta t)$ 
18:          end for
19:           $\mathbf{x}_t^{\text{pred}} \leftarrow \mathbf{x}_{t-1}^{\text{pred}} + \sum_{i=1}^k \alpha_i^{*(f)} G_i(\cdot)$ 
20:           $L \leftarrow L + \|\mathbf{x}_t^{\text{pred}} - \mathbf{x}_t^{\text{gt}}\|^2$ 
21:        end for
22:      end for
23:    end for
24:  end for
25:   $\theta = \{\theta_1, \dots, \theta_k\} \leftarrow \theta - \alpha \nabla_{\theta} L$ 
26: end while

```

multi-step loss encourages accurate long-term predictions. After convergence, the basis functions are frozen and ready for online adaptation.

2) *Online Adaptation:* During deployment, VA continuously adapts the coefficients α every 5 seconds using recent trajectories. This continuous adaptation mechanism enables VA to maintain accurate kinodynamics as terrain conditions evolve during navigation. The adapted coefficients are immediately used for forward prediction in the MPPI system, enabling real-time response to evolving terrain while preserving the generalization benefits of the pre-trained basis functions.

IV. IMPLEMENTATIONS

In this section, we describe the implementation details of dataset collection, preprocessing, the kinodynamic model architecture within the VA framework, and other baselines designed for adaptation to vertically challenging terrain.

A. Datasets and Preprocessing

Based on the Verti-Bench [35] simulator, we construct 100 off-road environments, each spanning 129m \times 129m with

TABLE I: Hyperparameters for VA

Parameter	Value
Model Architecture	
Number of basis functions (k)	24
Hidden dimension	$\lfloor 64\sqrt{k} \rfloor = 313$
Activation function	ReLU
Output dimension	6
Training Configuration	
Learning rate	10^{-3}
Optimizer	Adam
Learning rate scheduler	Cosine annealing
Total training steps	1000
Data Sampling	
Environments per batch (F)	5
Trajectories per environment (N)	10
Example trajectories (N_{ex})	4
Query trajectories (N_{q})	6
Sequences per trajectory (S)	2
Prediction horizon (T_{pred})	8
Integration	
RK4 time step (Δt)	0.1
Example batch size	256
Regularization (λ)	10^{-3}

0.1m pixel resolution. The environments are evenly distributed across low, medium, and high elevation levels. Each environment contains seven rigid (grass, wood, gravel, dirt, clay, rock, and concrete) and three deformable terrain semantic classes (snow, mud, and sand). Rigid terrain is modeled with a normal distribution of friction coefficients and a fixed restitution coefficient of 0.01, while deformable terrain is assigned physics parameters (cohesive effect, soil stiffness, and hardening effect) at three levels of deformability: soft, medium, and hard.

Elevation and RGB semantic patches are extracted from a 128 \times 128 pixel region beneath and aligned with the vehicle at 10 Hz. Elevation values are centered at the vehicle's current altitude. For feature extraction, we employ the SWAE generative model to embed elevation and semantic patches into a 64-dimensional latent space, preserving both geometric and semantic information for downstream kinodynamic modeling.

Since the mobile robot's kinodynamics are invariant under translation and rotation, we adopt a gravity-aligned body frame to enhance data efficiency and improve model accuracy. For each training sample, current state is denoted as: $[0, 0, 0, \phi_t, \varphi_t, 0, \mathbf{e}_{\text{elev}}, \mathbf{e}_{\text{sem}}] \in \mathbb{R}^{22}$, where the position components (x, y, z) and yaw ψ are zeroed (and therefore omitted), while roll ϕ_t and pitch φ_t retain their world-frame values. The terrain embeddings $\mathbf{e}_{\text{elev}} \in \mathbb{R}^8$ and $\mathbf{e}_{\text{sem}} \in \mathbb{R}^8$ are derived from the SWAE encoder and further compressed by a 3-layer MLP. The corresponding next state represents the change in vehicle pose: $[\Delta x, \Delta y, \Delta z, \phi_{t+1}, \varphi_{t+1}, \Delta \psi, \mathbf{e}_{\text{elev}}, \mathbf{e}_{\text{sem}}] \in \mathbb{R}^{22}$, where $(\Delta x, \Delta y, \Delta z, \Delta \psi)$ represent the body-frame relative motion, and $(\phi_{t+1}, \varphi_{t+1})$ are the absolute roll and pitch angles at the next timestep. We omit \mathbf{e}_{elev} and \mathbf{e}_{sem} in the prediction since

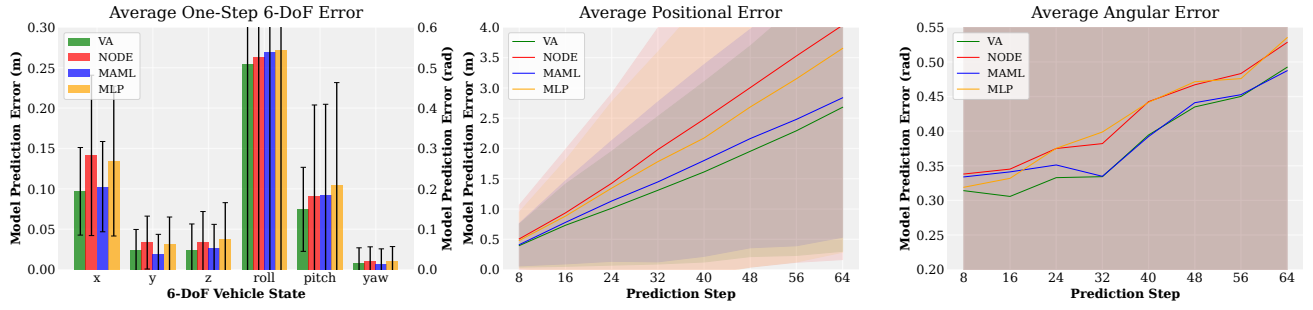


Fig. 3: Model Prediction Error of VA and three Baselines: Average One-Step 6-DoF Positional and Angular Error (Left); Prediction Error vs. Prediction Step (Middle and Right).

they can be acquired from perception.

The vehicle executes sinusoidal random exploration across all environments to collect training data. Steering follows $\mathbf{u}_{\text{steer}}(t) = \sin(\omega_s t)$ with $\omega_s \sim \mathcal{U}(0.1, 0.5)$ Hz, combined with speed $\mathbf{u}_{\text{speed}}(t) = v_c + A \sin(\omega_v t)$, where $\omega_v \sim \mathcal{U}(0.1, 2.5)$ Hz. The speed amplitude A and center velocity v_c are determined such that the minimum speed lies in $\mathcal{U}(0, 2)$ m/s and maximum speed is 3 m/s. This exploration strategy ensures diverse vehicle-terrain interactions, providing rich datasets for training the VA framework.

B. Model Architecture

Each neural ODE basis function g_i is a 4-layer MLP with $\{24, 313, 313, 313, 313, 6\}$ neurons and ReLU activations. The network takes a concatenated state-action vector as input and outputs a 6-DoF state derivative. The hidden dimension is scaled with the number of basis functions as $h = \lfloor 64\sqrt{k} \rfloor$ to balance model capacity and computational cost. For coefficient computation, we employ a memory-efficient batched least-squares solver that processes example sets in mini-batches of size 256. This design enables handling large datasets while maintaining numerical stability. A regularization term of $\lambda = 10^{-3}$ is applied to improve the numerical stability of the Gram matrix inverse in Eqn. (1).

The model is trained using multi-step rollout with gradient accumulation to balance memory efficiency and training stability. As shown in Table I, the cosine annealing schedule gradually decays the learning rate from 10^{-3} to 10^{-5} over 1000 training steps, promoting stable convergence and fine-grained parameter updates.

C. Baselines

VA is compared against three baseline methods: MLP, first-order MAML, and Neural ODE (NODE).

MLP employs a standard feedforward neural network with the same 4-layer architecture as VA’s basis functions. For online adaptation, we fine-tune only the last layer to reduce computational overhead while retaining adaptation capability. Given example data from a new terrain, the model performs 40,000 gradient steps until convergence using the Adam optimizer with a learning rate 5×10^{-3} , minimizing the Mean Squared Error (MSE) between predicted and ground-truth state changes until convergence.

First-order MAML is designed to learn initialization parameters that enable rapid adaptation. During offline training, the model learns across diverse terrain distributions using the standard MAML objective, allowing it to quickly adapt to new conditions. For adaptation, the meta-learned parameters are fine-tuned for 20,000 gradient steps to converge with an inner learning rate 5×10^{-3} . However, gradient-based updates remain computationally expensive and too slow to support real-time model inference in navigation systems such as MPPI.

To overcome this challenge, Neural ODE shares the same network architecture but integrates its outputs using the RK4 numerical method. By modeling kinodynamics in continuous time, Neural ODE provides smoother state transitions and more stable predictions. During offline training, the model learns to predict state derivatives that are integrated over a fixed time step of $\Delta t = 0.1$ s. For online adaptation, all network parameters are fine-tuned with only 500 gradient steps to converge. The reduced adaptation steps, compared to MLP and MAML, reflect the improved sample efficiency from continuous-time kinodynamic modeling. However, the method still relies on iterative optimization, limiting real-time performance, while VA only uses a least-squares solver.

V. EXPERIMENTS

We compare kinodynamics accuracy and simulated navigation performance in Verti-Bench [35], contrasting VA against the three baselines. Additionally, we compare kinodynamic predictions of VA and MAML using real-world data collected from a physical 1/10th-scale open-source Verti-4-Wheeler robot [5] in an off-road testbed.

A. Prediction Accuracy

We evaluate the accuracy of VA against all baselines in unseen off-road environments with high elevation level. As shown in Fig. 3 left, VA achieves the lowest one-step prediction errors for most 6-DoF state components, with an overall mean error of 0.1367 compared to Neural ODE (0.1561), MAML (0.1476), and MLP (0.1626). In particular, VA outperforms all baselines in rotational states, reducing roll and pitch error by 3.2% and 22.3% compared to Neural ODE. For translational states, VA achieves competitive accuracy, with the lowest error in the x direction (0.0969) and

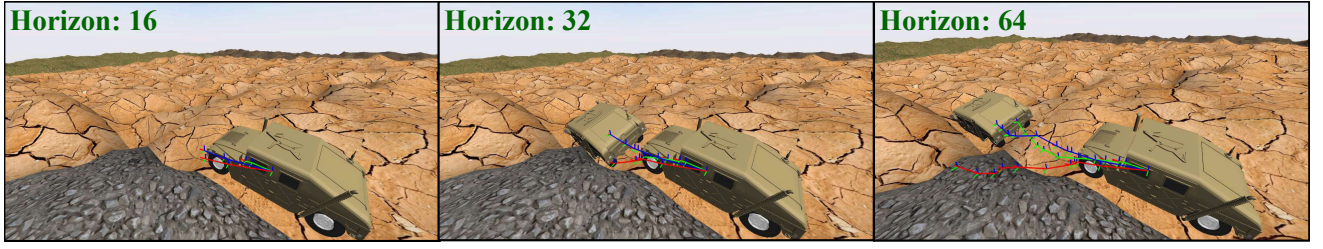


Fig. 4: 6-DoF Vehicle Trajectories of **VA**, **MAML**, and **Ground Truth** with Increasing Horizon: **VA** matches **Ground Truth** even with a long horizon, while **MAML** diverges more.

z direction (0.0243) relative to the baselines. These results indicate that VA is effective at modeling complex rotational dynamics while remaining robust for translational motions.

Moreover, VA demonstrates lower prediction variance, reflecting more consistent and reliable performance across diverse terrain conditions. As shown in Fig. 3 middle and right, when predictions are rolled out over time, all baselines experience rapid error growth and increasing uncertainty, especially for positional states. In contrast, VA maintains stable and accurate long-term predictions, with slower error accumulation and smaller uncertainty bounds.

Building on these findings, we further evaluate VA’s performance across different terrain difficulty levels. The results in Table II compare the proposed VA framework with three baselines, Neural ODE, MAML, and MLP, across low, medium, and high elevation levels.

TABLE II: Baseline evaluations across three terrain difficulty levels.

Method	MSE ↓			Adaptation Time ↓		
	Low	Medium	High	Low	Medium	High
VA (Proposed)	0.177	0.760	2.161	0.309	0.306	0.311
Neural ODE	0.200	0.999	3.128	1.715	1.737	1.734
MAML	0.186	0.814	2.093	11.175	11.366	10.616
MLP	0.214	0.904	2.214	40.889	40.556	40.777

The VA demonstrates superior performance in both accuracy and efficiency. In terms of prediction accuracy, VA achieves the lowest MSE on low (0.177) and medium (0.760) elevation levels, outperforming Neural ODE by 11.5% and 23.9%, MAML by 4.8% and 6.6%, and MLP by 17.3% and 15.9%, respectively. On high elevation level, VA achieves competitive performance (2.161) with only a 3.2% higher MSE compared to the best-performing MAML (2.093), while still outperforming Neural ODE by 30.9% and MLP by 2.4%. These results highlight VA’s consistent ability to model complex vehicle-terrain interactions across varying terrain difficulty levels.

In terms of adaptation time, VA adapts the fastest, requiring approximately 0.31s, which is over 5X faster than Neural ODE (~ 1.73 s) and more than 30X faster than MAML (~ 11 s) and MLP (~ 40 s). This highlights VA’s ability to

achieve both high accuracy and rapid adaptation, effectively balancing precision and efficiency.

Fig. 4 compares the predicted trajectories from VA and MAML against the ground truth across different prediction horizons in the most challenging high elevation terrain. At horizon 16, both methods closely follow the ground truth. By horizon 32, while both VA and MAML generally maintain the correct direction, MAML begins to drift to the left and VA slightly to the right. By horizon 64, MAML accumulates substantial error, with its trajectory extending onto rocky terrain on the left. In contrast, VA continues to closely track the ground truth.

B. Ablation Studies

We conduct ablation studies to evaluate the contribution of each key component in VA: (i) elevation embeddings and (ii) semantic embeddings. Each variant removes one of these components from the full VA framework:

- **VA**: The complete VA framework;
- **VA w/o semantic**: Removes the semantic embeddings. The model uses only the 6-DoF pose and elevation information, capturing geometric terrain features but lacking knowledge about terrain semantics such as friction or deformability;
- **VA w/o elevation**: Removes the elevation embeddings. In this variant, the model relies solely on semantic embeddings and the 6-DoF pose to capture terrain interaction, without direct geometric information about terrain elevation changes; and
- **VA w/o both**: Uses only the 6-DoF pose without elevation or semantic embeddings. This baseline captures purely kinodynamic behavior without terrain context, relying solely on vehicle motion states.

TABLE III: Ablation Study: VA MSE across three terrain difficulty levels.

Variant	MSE ↓		
	Low	Medium	High
VA (Proposed)	0.177	0.760	2.161
VA w/o semantic	0.201	0.761	2.292
VA w/o elevation	0.220	0.840	2.559
VA w/o both	0.177	0.761	3.038

The results in Table III underscore the importance of both elevation and semantic embeddings within the VA framework across terrain difficulty levels. On low elevation terrain, the complete VA framework achieves the lowest MSE (0.177), outperforming variants w/o semantic (0.201) and elevation (0.220) embeddings. Interestingly, the VA w/o both also achieves an MSE of 0.177, indicating that for low elevation level, basic kinodynamic modeling using only the 6-DoF pose is sufficient to achieve comparable accuracy. As terrain complexity increases, the impact of elevation embeddings becomes more pronounced. On medium elevation terrain, removing elevation information increases MSE by 10.5%, while removing semantic embeddings causes only a negligible 0.1% increase, showing that elevation data is key for capturing geometric variations. On high elevation terrain, all ablations show clear performance degradation: removing semantic embeddings increases error by 6.1%, while removing elevation embeddings causes a much larger 18.4% increase. The VA w/o both suffers the most, with a 40.5% increase in error compared to the full VA framework. These findings demonstrate that elevation embeddings are critical for geometric terrain understanding, semantic embeddings provide complementary insights for surface properties, and their combination is essential for robust vehicle-terrain interaction modeling in challenging off-road environments.

C. Simulated Navigation Performance

VA and the best baseline MAML are integrated into the MPPI planner, whose navigation performance is evaluated across terrain difficulty levels using three metrics: (i) number of successful trials (out of 5), (ii) mean traversal time (of successful trials in seconds), and (iii) average roll/pitch angles with variance (in degrees). We also compare VA adapted once per new environment (VA^{*}) and adapted every 5s (VA[†]), as well as MAML adapted once (MAML^{*}), since the significant computation disallows MAML do adapt online.

TABLE IV: **Simulated Navigation Performance:** Success Rate, Traversal Time, Roll, and Pitch. *: adapted once, †: adapted every 5s.

Low Elevation	VA [*]	VA [†]	MAML [*]
Success Rate ↑	5/5	5/5	4/5
Traversal Time ↓	31.16s	34.73s	28.90s
Roll ↓	2.36°±0.06°	2.23°±0.05°	2.61°±0.11°
Pitch ↓	2.17°±0.04°	1.98°±0.04°	1.97°±0.04°
Medium Elevation	VA [*]	VA [†]	MAML [*]
Success Rate ↑	3/5	2/5	0/5
Traversal Time ↓	33.94s	39.98s	-
Roll ↓	5.73°±0.25°	5.06°±0.62°	-
Pitch ↓	4.74°±0.07°	4.17°±0°	-
High Elevation	VA [*]	VA [†]	MAML [*]
Success Rate ↑	4/5	1/5	0/5
Traversal Time ↓	32.75s	32.29s	-
Roll ↓	9.18°±1.34°	5.83°±-	-
Pitch ↓	5.91°±0.17°	6.24°±-	-

Table IV shows that our single adaptation method, VA^{*}, achieves the best overall performance, consistently surpassing MAML^{*} in success rate. Surprisingly, VA[†] performs worse for medium and high elevation levels than VA^{*}. This is because VA^{*} computes a single set of coefficients that represent an averaged kinodynamic model after seeing data sampled from the entire new environment. However, VA[†] continuously updates its coefficients based on recent trajectory history, which can lead to mismatch when terrain types change significantly. As a result, the updated coefficients may work well for the previous terrain but become poorly suited when the vehicle encounters entirely new semantic terrain types. These findings indicate that, while our one-time adaptation strategy provides greater robustness and reliability, future work should focus on refining recursive updates to better handle rapid and frequent terrain changes.

D. Physical Experiment for Sim-to-Real Adaptation

Trained with simulation data from Verti-Bench [35], we adapt VA and MAML using physical data collected on real-world, vertically challenging, off-road terrain to test their sim-to-real adaptation. The real-world dataset includes driving a Verti-4-Wheeler robot through diverse rocky terrain in the off-road testbed, as shown in Fig. 1, for 70.3 seconds. We use 28.1 seconds of data for adaptation and evaluate and compare the kinodynamic modeling accuracy of VA and MAML with the rest 42.2 seconds of data.

TABLE V: Physical Experiment Results

Method	MSE ↓					
	x	y	z	roll	pitch	yaw
VA (Proposed)	0.014	0.010	0.002	0.066	0.041	0.024
MAML	0.016	0.201	0.003	0.068	0.059	0.025

The physical experiment results in Table V demonstrate that VA achieves more accurate predictions than MAML across all DoFs, especially in lateral motion. In particular, VA significantly reduces the y-direction error by 95% from 0.201 to 0.010, indicating a much stronger ability to model lateral vehicle kinodynamics on vertically challenging terrain. Both methods perform similarly in vertical motion (z) and yaw, but VA achieves lower errors in translational states (x, y) and orientation states (roll, pitch). These results confirm that VA can enable terrain-aware, sim-to-real adaptation, improves real-world prediction accuracy, and provides a more reliable foundation for navigation in complex off-road environments.

VI. CONCLUSIONS AND LIMITATIONS

We present VA, a novel online adaptation framework for terrain-aware kinodynamic modeling in off-road environments. By integrating elevation and semantic embeddings, VA enables rapid adaptation to unseen terrain through efficient least-squares optimization. Our evaluation demonstrates that VA improves prediction accuracy by up to 23.9% while

achieving 5X faster adaptation than baselines, effectively capturing complex vehicle-terrain interactions.

A key limitation of this work is that our adaptation method updates coefficients once, which may restrict its ability to handle rapidly changing terrain conditions. In future work, we plan to explore more robust recursive approaches, such as recursive least squares, to enable continuous and stable updates. In addition, elevation and semantic embeddings are currently combined through simple concatenation, which may not fully capture rich cross-modal relationships. To address this, we aim to investigate advanced fusion techniques, such as attention mechanisms or cross-modal transformers, to better integrate geometric and semantic information and further enhance adaptation performance.

REFERENCES

- [1] C. Min, S. Si, X. Wang, H. Xue, W. Jiang, Y. Liu, J. Wang, Q. Zhu, Q. Zhu, L. Luo *et al.*, “Autonomous driving in unstructured environments: How far have we come?” *arXiv preprint arXiv:2410.07701*, 2024.
- [2] N. Wang, X. Li, K. Zhang, J. Wang, and D. Xie, “A survey on path planning for autonomous ground vehicles in unstructured environments,” *Machines*, vol. 12, no. 1, p. 31, 2024.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] X. Cai, J. Queeney, T. Xu, A. Datar, C. Pan, M. Miller, A. Flather, P. R. Osteen, N. Roy, X. Xiao *et al.*, “Pietra: Physics-informed evidential learning for traversing out-of-distribution terrain,” *IEEE Robotics and Automation Letters*, 2025.
- [7] C. Pan, A. Datar, A. Pokhrel, M. Choulas, M. Nazeri, and X. Xiao, “Traverse the non-traversable: Estimating traversability for wheeled mobility on vertically challenging terrain,” *arXiv preprint arXiv:2409.17479*, 2024.
- [8] K. S. Sikand, S. Rabiee, A. Uccello, X. Xiao, G. Warnell, and J. Biswas, “Visual representation learning for preference-aware path planning,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 11 303–11 309.
- [9] 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*, vol. 9, no. 11, pp. 9653–9660, 2024.
- [10] 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.
- [11] I. Clavera, J. Rothfuss, J. Schulman, Y. Fujita, T. Asfour, and P. Abbeel, “Model-based reinforcement learning via meta-policy optimization,” in *Conference on robot learning*. PMLR, 2018, pp. 617–629.
- [12] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, “Neural ordinary differential equations,” *Advances in neural information processing systems*, vol. 31, 2018.
- [13] T. Ingebrand, A. Thorpe, and U. Topcu, “Zero-shot transfer of neural odes,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 67 604–67 626, 2024.
- [14] 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.
- [15] A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn, “Learning to adapt in dynamic, real-world environments through meta-reinforcement learning,” *arXiv preprint arXiv:1803.11347*, 2018.
- [16] S. M. LaValle and J. J. Kuffner Jr, “Randomized kinodynamic planning,” *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] 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. Sindhwan, “Learning model predictive controllers with real-time attention for real-world navigation,” in *Conference on robot learning*. PMLR, 2022.
- [30] C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, “Online meta-learning,” in *International conference on machine learning*. PMLR, 2019, pp. 1920–1930.
- [31] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, “Meta-learning with latent embedding optimization,” *arXiv preprint arXiv:1807.05960*, 2018.
- [32] B. Liu, X. Xiao, and P. Stone, “A lifelong learning approach to mobile robot navigation,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1090–1096, 2021.
- [33] G. Williams, A. Aldrich, and E. A. Theodorou, “Model predictive path integral control: From theory to parallel computation,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 344–357, 2017.
- [34] D. Tan and Z. Chen, “On a general formula of fourth order runge-kutta method,” *Journal of Mathematical Science & Mathematics Education*, vol. 7, no. 2, pp. 1–10, 2012.
- [35] T. Xu, C. Pan, M. B. Rao, A. Datar, A. Pokhrel, Y. Lu, and X. Xiao, “Verti-bench: A general and scalable off-road mobility benchmark for vertically challenging terrain,” in *Robotics: Science and Systems (RSS)* 2025, 2025.