2025 IEEE International Conference on Robotics and Automation (ICRA)
May 19-23, 2025. Atlanta, USA

2025 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3315-4139-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICRA55743.2025.11128598

VertiCoder: Self-Supervised Kinodynamic Representation Learning on Vertically Challenging Terrain

Mohammad Nazeri[1], Aniket Datar[1], Anuj Pokhrel[1], Chenhui Pan[1], Garrett Warnell[2,3], and Xuesu Xiao[1]

***Abstract*— We present VertiCoder, a self-supervised representation learning approach for robot mobility on vertically challenging terrain. Using the same pre-training process, VertiCoder can handle four different downstream tasks, including forward kinodynamics learning, inverse kinodynamics learning, behavior cloning, and patch reconstruction with a single representation. VertiCoder uses a TransformerEncoder to learn the local context of its surroundings by random masking and next patch reconstruction. We show that VertiCoder achieves better performance across all four different tasks compared to specialized End-to-End models with 77% fewer parameters. We also show VertiCoder's comparable performance against state-of-the-art kinodynamic modeling and planning approaches in real-world robot deployment. These results underscore the efficacy of VertiCoder in mitigating overfitting and fostering more robust generalization across diverse environmental contexts and downstream vehicle kinodynamic tasks[1].**

I. Introduction

Wheeled robots, commonly employed in structured environments such as warehouses, homes, and offices, often encounter limitations when navigating off-road terrain characterized by vertical challenges [1]. In applications like search and rescue and remote inspection, wheeled robots frequently face rocky or uneven terrain filled with obstacles of similar size as the robots. Their inability to negotiate through such vertical protrusions from the ground can disrupt robots' stability, cause wheel slippage, and even lead to damage. Such a limitation presents a key obstacle in expanding the operational domain of wheeled robots from controlled settings to unstructured off-road environments.

Recent advancements in wheeled mobility have demonstrated the feasibility of navigating vertically challenging terrain with minimal hardware modifications. Despite the complexity of the terrain, simple all-wheel drive, independent suspensions, and differential locking have proven sufficient in enabling wheeled robots to traverse such environments [1]. Concurrent strides in data-driven robot perception, motion planning, and vehicle control have further contributed to these achievements [2]–[4]. Among all these successes in traversing vertically challenging terrain [1]–[3], accurate kinodynamic understanding plays a vital role in enabling safe and efficient robot mobility, ranging from forward and inverse kinodynamic modeling [5]–[8] as well as behavior cloning based on geometric terrain input [9], [10].

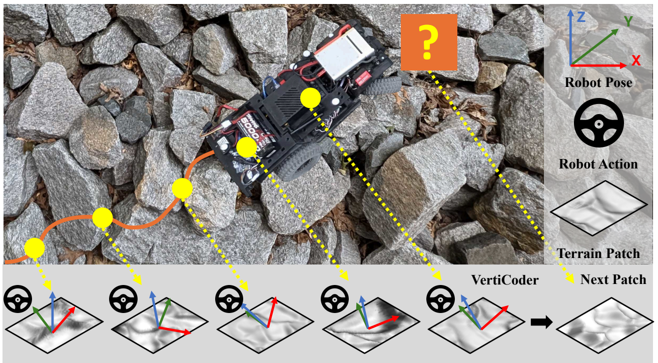

Fig. 1: VertiCoder uses a Transformer to encode a history of terrain patches, robot actions, and robot poses and predicts the representation of the next patch. The learned representation is used in multiple downstream tasks.

While those end-to-end learning approaches have shown promise, their task-specific nature limits their generalizability to diverse scenarios. In contrast, recent breakthroughs in Self-Supervised Learning (SSL) [11] suggest the possibility of achieving zero-shot or few-shot learning across various tasks [12], [13]. For wheeled mobility on vertically challenging terrain, learning a robust and general terrain representation to facilitate kinodynamic understanding will simplify or even improve learning any mobility-related downstream tasks like behavior cloning and 6-DoF kinodynamic modeling. Therefore, in this work, we ask the question: *How can we develop an accurate and generalizable terrain representation for kinodynamic understanding capable of performing multiple tasks with comparable accuracy to its end-to-end counterparts?* Given the successes of SSL in robotics for solving multiple manipulation tasks [14], [15], we introduce VertiCoder, an SSL model that utilizes a transformer encoder [16] to reconstruct masked tokens and predict an accurate representation of future terrain patches underneath the robot in vertically challenging terrain. Inspired by VisionTransformer (ViT) [17] and VANP [18], VertiCoder leverages a TransformerEncoder with an additional context token to reconstruct the masked tokens and next patch underneath the robot. This approach boosts robots' local context awareness. Subsequently, downstream tasks can leverage this context token, which contains historical information of terrain patches, robot poses, and robot actions (Fig. 1), to perform forward kinodynamics learning (FKD), inverse kinodynamics learning (IKD), behavior cloning (BC), and patch reconstruction (PR).

[1]Department of Computer Science, George Mason University {mnazerir, adatar, apokhre, cpan7, xiao}@gmu.edu [2]DEVCOM Army Research Laboratory garrett.a.warnell.civ@army.mil [3]Department of Computer Science, The University of Texas at Austin

[1] https://github.com/mhnazeri/VertiCoder

979-8-3315-4139-2/25/$31.00 ©2025 IEEE

Our experimental results demonstrate better generalization on unseen test environments compared with specialized End-to-End (E2E) models and comparable performance in real-world robot deployment against state-of-the-art models. The contributions of this work can be summarized as follows:

- An SSL model for kinodynamic understanding that can accurately predict future terrain representations;
- four different mobility-related downstream tasks from one representation; and
- real-world comparison with previous methods in terms of both learning and robot deployment performance.

II. Related Work

In this section, we first discuss the current approaches in data-driven kinodynamics learning, followed by an exploration of the recent advancements in applications of self-supervised learning in robotics.

Learning robot kinodynamics is a crucial step toward achieving successful navigation in off-road environments [4]. While terrain characteristics [19] and robot reaction to such terrain can often be predicted [20], [21] to some extent, the inherent complexity and variability [22] of off-road terrain make it difficult for robots to attain the commanded velocities [23]. To address these challenges, researchers have increasingly focused on learning robot kinodynamics in off-road environments to develop robust control strategies [24] that account for uncertainties and challenges posed by varying terrain conditions [25], [26]. This enables robots to adapt their behavior in real-time during high-speed navigation [6]–[8], jumping over small hills [5], mitigating rollovers [27], [28], or even traversing through vertically challenging terrain [1]–[3]. By learning the robot kinodynamics, researchers have also enabled robots to navigate off-road environments in a risk-aware manner [29]–[31]. Given the data-intensive nature of many kinodynamics learning methods, recent research has increasingly explored physics-informed [32]–[34], and self-supervised approaches [3], [31] to reduce the reliance on extensive datasets. VertiCoder is a self-supervised kinodynamics learning approach that aims at efficiently improving multiple downstream kinodynamic tasks.

Self-Supervised Learning (SSL) for robotics has emerged as a valuable technique for reducing the reliance on labeled data and improving generalization to unseen environments, a common limitation in supervised learning methods. SSL techniques leverage intrinsic signals within the data to guide machine learning models, broadly categorized into contrastive [35], [36] and information maximization methods [31]. SSL methods have been successfully applied to a variety of off-road robotics tasks, including learning terrain properties from sensor readings such as force sensors [37] and IMU [38], estimating traversability using vision [36], [39]–[42], pointclouds [43], [44], or multi-modal sensor data [20], and acquiring terrain preferences [45]–[47]. Learning terrain representation [3], [31], [48] is another application of SSL, where the distilled representation is learned based on the property of the terrain which can be then used to perform various downstream tasks in off-road navigation. TAL [3], one of the closest works to VertiCoder, leverages the entire elevation map and robot pose to reconstruct the terrain patch beneath the robot as a pretext task for learning terrain representations. However, this reliance on a complete map during pre-training can be limiting, as such maps are not always available in real-world scenarios. VertiCoder, in contrast, employs masking and next-patch prediction as its pretext task, thereby eliminating the need for a full map. This approach not only enhances the learned representation's generalizability but also broadens its applicability to a wider range of downstream tasks.

Although first designed for natural language processing, Transformers [16] have revolutionized various domains including computer vision [17], [49] and robotics [36], [50]. By leveraging the attention mechanism, Transformers can effectively learn meaningful representations [51], [52] from unlabeled data by predicting the next token or masking the surrounding tokens. Next token prediction has been applied in cross-embodied robot policy learning [53], [54]. The CrossFormer model [54] used diverse large datasets to learn a wide range of navigation and manipulation tasks for different robot embodiments. However, its context size of 2135 is considerably larger than the more manageable VertiCoder's context size of 61 for a mobile robot with limited onboard computational resources. Moreover, the network's size and complexity make CrossFormer unsuitable for real-time decision-making in vertically challenging environments and deployment on a small Verti-Wheeler platform [1].

III. Approach

Navigating through vertically challenging terrain presents unique challenges in terrain representation learning. In contrast to ego-centric visual navigation, we cannot easily anticipate future obstacles, which complicates decision-making and increases the risk of encountering unforeseen traps. In this section, we first define the pretext task to train the VertiCoder and how it helps VertiCoder to anticipate future obstacles. Then we define FKD, IKD, BC, and PR as the downstream tasks (Fig. 2).

A. *Pre-training*

Next token prediction has shown great potential in Large Language Models as a pretext objective [55], [56]. Inspired by this approach, we propose leveraging the prediction of the next terrain patch underneath the robot, in conjunction with randomly masked token reconstruction, as a pretext task to align and relate patches with corresponding robot actions and poses. This pretext task facilitates an understanding of the surrounding context and enables the anticipation of future obscured obstacles, thereby enhancing performance and generalization to unseen environments on downstream tasks.

Architecture. VertiCoder leverages a TransformerEncoder, inspired by BERT and ViT. It incorporates an additional learnable token, `ctx`, analogous to BERT's CLS token. For patch encoding, VertiCoder employs a pre-trained

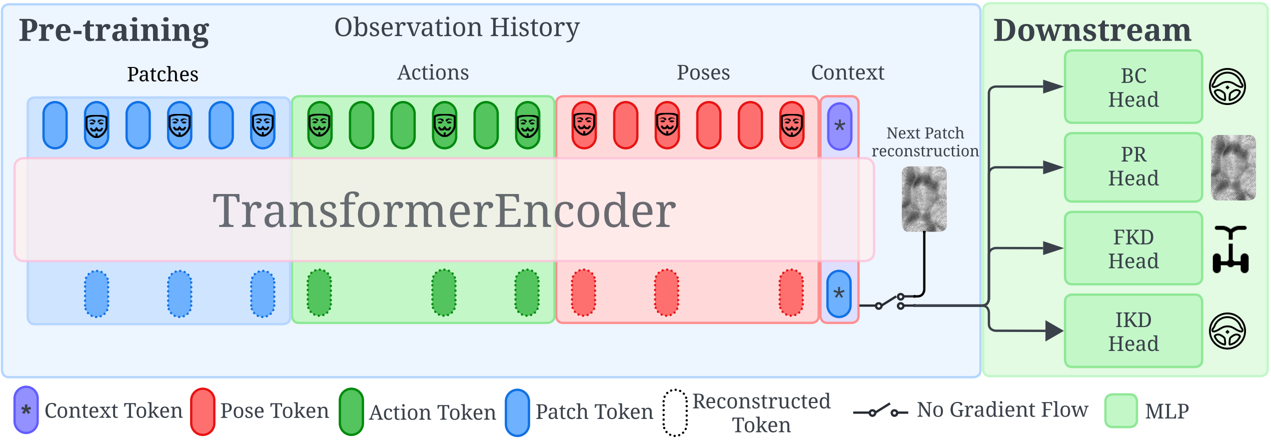

Fig. 2: **VERTICODER Architecture**. VERTICODER employs a TransformerEncoder with a context token to predict the representation of the next patch beneath the robot. It achieves this through two simultaneous pretext tasks: predicting the next patch in a sequence of patches, actions, and poses, and reconstructing randomly masked tokens (🎭) within that sequence. Subsequently, we freeze the VERTICODER and attach multiple downstream task heads to the context token, exclusively training the downstream task heads.

encoder from a SWAE [57], a component with 3.67 million parameters that constitutes the majority of VERTICODER's model size.

Train. To train VERTICODER, we use a dataset of $(\mathbf{i}_t, \mathbf{a}_t, \mathbf{p}_t) \in \mathcal{O}$, where $\mathbf{i}_t$ is a terrain patch of size 40×40 beneath the robot at time t, $\mathbf{a}_t$ is the action (linear (v_t) and angular (ω_t) velocities) taken by the robot on the patch $\mathbf{i}_t$, $\mathbf{p}_t$ is the robot pose in $\mathbb{SE}(3)$ on the patch $\mathbf{i}_t$, and $\mathcal{O}$ is the observation space. We pass a history of 20 timesteps as an input to VERTICODER. We use three separate encoders to map $(\mathbf{i}_t, \mathbf{a}_t, \mathbf{p}_t)$ to consequent tokens. For terrain patches, $\mathbf{i}_t$, we use the frozen SWAE encoder, denoted by SW_{enc}, to encode the patches into patch tokens $\tau^{\mathbf{i}}_t$. For actions $\mathbf{a}_t$, we employ a linear model, $\mathrm{enc}_{\mathbf{a}}$, to map actions into action tokens, denoted as $\tau^{\mathbf{a}}_t$. And finally, to tokenize robot poses $\mathbf{p}_t$, we use another linear model, $\mathrm{enc}_{\mathbf{p}}$, to map poses into pose tokens, denoted by $\tau^{\mathbf{P}}_t$. Then, we prefix the observation tokens $(\tau^{\mathbf{i}}_i, \tau^{\mathbf{a}}_i, \tau^{\mathbf{P}}_i)^t_{i=t-19}$ with a context token $\mathbf{ctx}$ and randomly masked 75% of the tokens before passing them to VERTICODER. The training of the VERTICODER is then accomplished by calculating and minimizing the mean squared error between the predicted and actual terrain patch embeddings.

B. *Downstream Tasks*

Forward Kinodynamics Learning (FKD): We follow the definition of Datar *et al.* [3] for FKD and adopt a discrete vehicle forward kinodynamic model:

$$\mathbf{p}_{t+1} = f_{\psi}(\mathbf{ctx}),$$

where $\mathbf{ctx}$ is VERTICODER's context token, $\mathbf{p}_{t+1}$ is the next robot pose, and ψ is learnable parameters. We aim to show that the learned representation possesses all the necessary information to predict the subsequent pose without explicitly taking in any other information. However, for the end-to-end model used for comparison, we enhance the input by explicitly incorporating the current action $\mathbf{a}_t$ and pose $\mathbf{p}_t$ in addition to the patch embedding.

The FKD model can be used in sampling-based motion planners to produce potential future vehicle trajectories, which will be evaluated based on a cost function, to move the robot to its goal while minimizing the chance of failure on vertically challenging terrain (e.g., rollover or getting stuck).

Inverse Kinodynamics Learning (IKD): We follow the definition of Karnan *et al.* [7] for IKD and use the context token as input to the vehicle inverse kinodynamic model:

$$\mathbf{a}_t = f_{\epsilon}(\mathbf{ctx}, \mathbf{p}_{t+1}),$$

where $\mathbf{a}_t$ is the current action to be taken by the robot in order to reach $\mathbf{p}_{t+1}$, the robot's desired next pose including translations and rotations along the x, y, and z axes. ϵ is learnable parameters.

The IKD model can be used with a global planner that constantly produces the robot's desired next state to move the robot toward its goal safely.

Behavior Cloning (BC): We follow the definition of Nazeri *et al.* [18], [58] for Behavior Cloning:

$$\mathbf{a}_t = \pi_{\zeta}(\mathbf{ctx}, g),$$

where π is a controller policy parameterized by ζ and g is a goal. If we do not include the goal, the policy learns to explore or drive forward without a goal, depending on the training data.

Patch Reconstruction (PR): To reconstruct the next patch, the SWAE decoder reconstructs the subsequent patch from the context token. This process is formulated as:

$$\mathbf{i}_{t+1} = \mathrm{SW}_{dec}(\mathbf{ctx}),$$

where $\mathbf{i}_{t+1}$ is the next terrain patch and SW_{dec} denotes the decoder (in contrast to the SWAE encoder SW_{enc} used in VERTICODER pre-training). Notice that the PR downstream task is also the same VERTICODER pre-training task.

All downstream tasks are learned by minimizing supervised losses between the prediction and the ground truth of next pose $\mathbf{p}_{t+1}$, current action $\mathbf{a}_t$, and next patch $\mathbf{i}_{t+1}$ for FKD, IKD, BC, and PR, respectively. For the PR task, we also use the peak signal-to-noise ratio (PSNR) metric to evaluate the quality of the reconstructed images.

C. Implementations

In terms of hardware configuration, this work employs an open-source Verti-4-Wheeler (V4W) platform, as described by Datar *et al.* [1]. The V4W is equipped with a Microsoft Azure Kinect RGB-D camera and an NVIDIA Jetson Xavier processor. Additionally, it features low-gear drive and lockable front and rear differentials, which significantly enhance its mobility on vertically challenging terrain. On the software side, VERTICODER is implemented with PyTorch and trained on a single A5000 GPU with 24GB memory.

Architecture: VERTICODER consists of a TransformerEncoder with an additional context vector like BERT [59], ViT [17], and VANP [18] with four layers and four heads to produce context embedding $\mathbf{ctx} \in \mathbb{R}^{128}$. The context size of the Transformer is 61 including 20 patches, 20 actions, 20 poses, and 1 context token. Using the Transformer along with masking and next patch prediction allows the model to understand the local context of the terrain and how a variation in terrain height and robot action can affect the robot pose over time. For downstream tasks, we use separate small multi-layer perceptrons with $[128, 64, 32, 16, n]$ layers, where n is the downstream output dimension to map the context token to the target task space.

Optimization: For both pretext and downstream training we use the AdamW optimizer [60]. We pre-train VERTICODER for 300 epochs with a batch size of 512. We then train the downstream heads for 50 epochs for both VERTICODER and its end-to-end counterpart with a batch size of 32 for a fair comparison. We observe that using a larger batch size than 32 for the downstream heads causes the model to converge to the mean of the data.

Freezing and Fine-tuning: We employ two versions of the training process for downstream tasks. First, we freeze the weights of the VERTICODER and train only the downstream task-specific heads. This strategy allows us to evaluate the expressivity and generalizability of VERTICODER's learned common representations. Second, in the fine-tuned version, we unfreeze the VERTICODER weights, allowing them to be updated in conjunction with the downstream heads using the downstream loss. This fine-tuning process is implemented to enhance performance, as it enables the VERTICODER model to adapt its pre-learned common representations to the specific nuances of each downstream task, potentially leading to more task-specific feature extraction and improved overall performance.

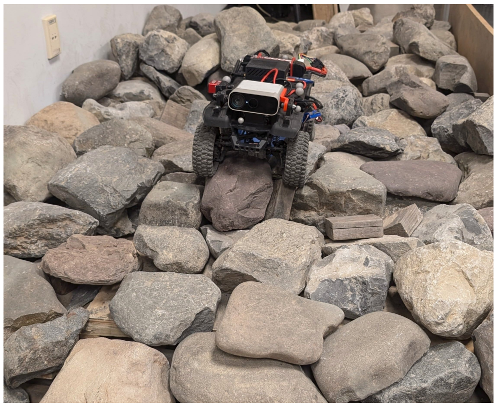

Fig. 3: **Rock Testbed and V4W used for Data Collection and Experiments**. The modularity of the testbed allows diverse rock configurations for training and evaluation.

End-to-End Architecture: For the specialized E2E models, we use Resnet-18 [61] as the patch encoder and attach multi-layer perceptrons with $[512, 256, 512, 64, n]$ layers as the task heads, where n is the dimension of downstream output. A notable distinction between these end-to-end models and the VERTICODER lies in the input modality where the former explicitly incorporates additional information, such as current pose and action to predict next pose for FKD and to predict next patch for PR, whereas the latter exclusively utilizes the context token $\mathbf{ctx}$, which implicitly includes such information through pre-training, as input to the downstream heads. A comparison of VERTICODER against such end-to-end architectures with explicit task-specific inputs aims to showcase the effectiveness of VERTICODER in extracting relevant and common features to implicitly use them for different downstream tasks during the same pre-training process.

Dataset: We use the dataset employed by TAL [3], which is collected on a rock testbed measuring 3.1 m × 1.3 m with a maximum height of 0.6 m (Fig. 3). The rock testbed's modular design allows for easy reconfiguration, facilitating mobility experiments across various terrain arrangements. Given that ground vehicle dynamics are predominantly influenced by terrain topology, and considering the substantial computational requirements associated with full 3D mapping, a 2.5D terrain elevation map is employed to construct the patches beneath the vehicle [62]. The dataset includes 30 minutes of teleoperating the robot on the rock testbed and an additional 30 minutes on a planar surface. The dataset is divided using a 9:1 ratio for training and testing purposes. The dataset encompasses visual inertial odometry for vehicle state estimation, elevation maps generated from depth images, and teleoperated vehicle control inputs, including throttle and steering commands. The rock testbed dataset captures a diverse range of 6-DoF vehicle states, including instances of vehicle rollover and immobilization.

IV. EXPERIMENTS

To validate VERTICODER's efficacy on the four specified downstream tasks while relying solely on the context token,

TABLE I: **Downstream Performance.** Comparison of the VertiCoder performance on four different tasks compared with end-to-end models on seen/unseen data. ❄ denotes frozen VertiCoder backbone, and 🔥 denotes fine-tuned model.

Method	**#Params (M)** ↓		**FKD** ↓		**IKD** ↓		**BC** ↓		**PR** ↑
	Learnable	Total	Train	Test	Train	Test	Train	Test	Test
EndToEnd-18	≈ 11.47	≈ 11.47	**0.002**	0.031	**0.012**	0.048	**0.015**	0.259	18.91
EndToEnd-50	≈ 26.42	≈ 26.42	0.002	0.031	0.039	0.117	0.017	0.292	-
EndToEnd-EfficientB4	≈ 21.04	≈ 21.04	0.010	0.032	0.037	0.120	0.108	0.283	-
VertiCoder ❄	**0.01**	**2.71**	0.007	**0.009**	0.073	**0.045**	0.172	**0.159**	**25.65**
VertiCoder 🔥	0.88	2.71	0.003	0.002	0.008	0.002	0.032	0.017	28.093

TABLE II: **Real-World Robot Deployment.** Comparison between VertiCoder BC and E2E BC.

	VertiCoder BC	E2E BC
Success Rate ↑	**8/10**	7/10
Average Time ↓	14.47±1.81	**12.28±2.69**

TABLE III: **Mask Percentage Ablation Study.** Comparison of different percentages of masking on FKD and PR.

Task Mask	**FKD** ↓		**PR** ↑
	Train	Test	Test
75%	0.058	**0.050**	**22.775**
90%	**0.006**	0.055	16.623

we conduct a comparative analysis between VertiCoder's predictions and those of specialized E2E models. Furthermore, we demonstrate the practical applicability of VertiCoder by deploying it on the V4W robot to navigate through vertically challenging terrain using BC. This dual approach of comparative evaluation and practical implementation serves to comprehensively assess the versatility and performance of the VertiCoder framework across diverse task domains.

A. VertiCoder *Experiments*

Accuracy across Different Downstream Tasks: A comparative analysis of VertiCoder's accuracy across four different downstream tasks, FKD, IKD, BC, and PR, against specialized E2E models reveals noteworthy findings, as illustrated in Table I. E2E-18, E2E-50, and E2E-EfficientB4 leverage Resnet-18, Resnet-50, and EfficientNet-B4 as their backbone respectively. The E2E models demonstrate a tendency to overfit the training data, resulting in suboptimal performance when confronted with unseen data. Increased depth in E2E-50 and E2E-EfficientB4 does not help the model to understand the complexities and cause the model to overfit more than the shallower E2E-18 model In contrast, both the frozen and fine-tuned versions of VertiCoder exhibit superior generalization capabilities across tasks and unseen environments. The enhanced generalization of VertiCoder can be attributed to its use of masking and next token prediction as pretext tasks during the pre-training phase. Note that VertiCoder downstream heads only receive the context token as input while E2E models have access to additional information as mentioned in Sec. III. Since PR is also part of VertiCoder's pretext task and its downstream head, i.e., SW$_{dec}$, does not need to be re-trained, it does not have a column for training loss on seen data.

Furthermore, VertiCoder's superior resistance to overfitting is achieved using a mere 23% of the parameters employed by the E2E model. Notice that the difference between Learnable (0.88M) and Total (2.71M) parameters for the fine-tuned VertiCoder is due to the always frozen SW$_{enc}$ parameters derived from SWAE encoder and decoder pre-training prior to VertiCoder pre-training. This also shows that most of the VertiCoder's parameters are coming from SW$_{enc}$. This marked reduction in parameter count, coupled with improved generalization, underscores VertiCoder's efficient training and inference time. Moreover, this approach yields an additional benefit in the form of reduced training time for downstream tasks. The compact yet expressive representations learned by VertiCoder outperform or produce comparable results against four different specialized E2E models and facilitate more efficient fine-tuning, as the model requires less adaptation to new task-specific objectives.

On-robot deployment: We deploy the VertiCoder with BC on the V4W. We conduct a comparative analysis of VertiCoder's performance against E2E BC. We utilize the reported results from TAL as our baseline for comparison and like TAL [3], we set the goal across the rock test bed. As demonstrated in Table II, VertiCoder BC outperforms E2E BC in terms of a higher success rate with a slower, but more steady speed. However, we observe that VertiCoder BC exhibits a performance gap when compared to WMVCT [2] and TAL [3], two specialized kinodynamic modeling approaches used in conjunction with sophisticated sampling-based motion planners. This suggests that while VertiCoder offers promising results in the real world, there remains room for further enhancement to achieve parity with state-of-the-art methods like WMVCT and TAL. Future research could explore the integration of complementary techniques to further refine the representation and narrow this performance gap.

B. *Ablation Studies*

To investigate the most effective approach to train VertiCoder we experiment with a series of ablation studies.

TABLE IV: **Token Ordering Ablation Study.** Comparison of different ways to feed tokens to VERTICODER on FKD and PR.

Task / Ordering	FKD ↓		PR ↑
	Train	Test	Test
Interleaving	0.065	0.052	22.306
Sequential	**0.058**	**0.050**	**22.775**

We choose the FKD accuracy and PR PSNR accuracy for ablations. We collect a new 30-minute dataset on the rock testbed which is split into 9:1 ratio for the train and test data. The ablation dataset is collected using a motion capture system for odometry while keeping the rest of the inputs the same.

Role of Masking: To investigate the impact of masking on accuracy, we conduct experiments with varying masking percentages. As detailed in Table III, we evaluate the model's performance with masking percentages of 75% and 90%, in line with recommendations from previous studies [59], [63]. Our findings reveal that masking 90% of the tokens hinders the model's ability to effectively grasp the context while masking 75% of the tokens facilitates a more effective representation of the surrounding information.

Order of tokens: Furthermore, we explore the influence of token ordering within the representation. We experiment with both interleaved token arrangements, $\{\tau^{\mathbf{i}}_{t-19}, \tau^{\mathbf{a}}_{t-19}, \tau^{\mathbf{p}}_{t-19}, \dots, \tau^{\mathbf{i}}_{t}, \tau^{\mathbf{a}}_{t}, \tau^{\mathbf{p}}_{t}\}$, and sequential ordering, $\{\tau^{\mathbf{i}}_{t-19}, \dots, \tau^{\mathbf{i}}_{t}, \tau^{\mathbf{a}}_{t-19}, \dots, \tau^{\mathbf{a}}_{t}, \tau^{\mathbf{p}}_{t-19}, \dots, \tau^{\mathbf{p}}_{t}\}$. The results, presented in Table IV, demonstrate superior model performance in both FKD and PR tasks when employing sequential ordering. This suggests that a sequential arrangement of tokens contributes to enhanced alignment between the different modalities.

C. Discussions

The disparity in performance of E2E models and VERTICODER can be attributed to the distinct architectural and training paradigms employed by each model. The E2E models, while potentially excelling in capturing intricate patterns within the training set, appear to lack the robustness required for effective generalization. Conversely, VERTICODER's architecture coupled with its training methodology—involving both a frozen and unfrozen representation—seems to confer enhanced adaptability to unfamiliar scenarios. The frozen version of VERTICODER, leveraging pre-trained representations without task-specific adjustments, demonstrates the inherent transferability of its learned features. The fine-tuned variant further refines these representations, striking a balance between retaining generalizable knowledge and adapting to task-specific nuances. The VERTICODER's self-supervised learning technique appears to be instrumental in cultivating a more robust and generalizable kinodynamic representation of the underlying data distribution.

D. Limitations

In our analysis, we identify several key limitations of the VERTICODER. Although VERTICODER demonstrates superior performance on the test dataset compared to E2E models, it falls short of surpassing state-of-the-art models, WMVCT and TAL, in real-world scenarios. This indicates that while VERTICODER's representation helps prevent overfitting on the training data, it still lacks the expressivity required to generalize effectively to real-world situations. We attribute this limitation to VERTICODER's reliance on local context awareness (i.e., its dependence on the patch directly beneath the robot), which excludes crucial global information from the map. This restricted access to global context hampers VERTICODER's ability for long-horizon planning. Furthermore, when the robot moves at a slow pace, there is minimal change in information between consecutive terrain patches. This lack of temporal variation can hinder the model's learning process. While data augmentation can potentially alleviate this issue, crude augmentations on elevation maps risk significantly altering the context, leading to inaccurate predictions in downstream tasks. Therefore, it is essential to develop more sophisticated augmentation techniques that preserve contextual integrity while providing sufficient variation for effective learning.

V. CONCLUSIONS

In this paper, we introduce VERTICODER, a self-supervised model designed to learn a general kinodynamic representation via a TransformerEncoder that can be applied across four distinct tasks for robot mobility on vertically challenging terrain: forward kinodynamics learning, inverse kinodynamics learning, behavior cloning, and patch reconstruction. This versatility is achieved through a pretext task involving random masking and next-patch reconstruction, promoting an effective representation of the robot's local context. Our experiments demonstrate that VERTICODER not only outperforms specialized end-to-end models across all four tasks while utilizing 77% fewer parameters, but also exhibits comparable performance to state-of-the-art approaches in real-world deployments. These results highlight the effectiveness of VERTICODER's self-supervised approach in mitigating overfitting and facilitating robust generalization across diverse and challenging tasks and environments.

REFERENCES

[1] 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)*, 2024.

[2] A. Datar, C. Pan, and X. Xiao, "Learning to model and plan for wheeled mobility on vertically challenging terrain," *arXiv preprint arXiv:2306.11611*, 2023.

[3] A. Datar, C. Pan, M. Nazeri, A. Pokhrel, and X. Xiao, "Terrain-attentive learning for efficient 6-dof kinodynamic modeling on vertically challenging terrain," *arXiv preprint arXiv:2403.16419*, 2024.

[4] 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.

[5] H. Lee, T. Kim, J. Mun, and W. Lee, "Learning terrain-aware kinodynamic model for autonomous off-road rally driving with model predictive path integral control," *IEEE Robotics and Automation Letters*, 2023.

[6] 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*, 2021.

[7] 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.

[8] 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.

[9] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[10] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *IEEE International Conference on Robotics and Automation*. IEEE, 2017.

[11] R. Balestriero, M. Ibrahim, V. Sobal, A. S. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. G. Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsiavash, Y. LeCun, and M. Goldblum, "A cookbook of self-supervised learning," *ArXiv*, vol. abs/2304.12210, 2023.

[12] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6892–6903.

[13] R. Doshi, H. Walke, O. Mees, S. Dasari, and S. Levine, "Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation," *arXiv preprint arXiv:2408.11812*, 2024.

[14] V. Myers, B. C. Zheng, O. Mees, S. Levine, and K. Fang, "Policy adaptation via language optimization: Decomposing tasks for few-shot imitation," *arXiv preprint arXiv:2408.16228*, 2024.

[15] Z. Zhou, P. Atreya, A. Lee, H. Walke, O. Mees, and S. Levine, "Autonomous improvement of instruction following skills via foundation models," *arXiv preprint arXiv:2407.20635*, 2024.

[16] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[18] M. Nazeri, J. Wang, A. Payandeh, and X. Xiao, "VANP: Learning where to see for navigation with self-supervised vision-action pre-training," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024.

[19] S. Ghosh, K. Otsu, and M. Ono, "Probabilistic kinematic state estimation for motion planning of planetary rovers," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5148–5154.

[20] 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)*. IEEE, 2023, pp. 931–938.

[21] T. Han, S. Talia, R. Panicker, P. Shah, N. Jawale, and B. Boots, "Dynamics models in the aggressive off-road driving regime," *arXiv preprint arXiv:2405.16487*, 2024.

[22] J. Gibson, B. Vlahov, D. Fan, P. Spieler, D. Pastor, A.-a. Agha-mohammadi, and E. A. Theodorou, "A multi-step dynamics modeling framework for autonomous driving in multiple environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7959–7965.

[23] 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)*, 2021, pp. 4844–4850.

[24] E. Pagot, M. Piccinini, and F. Biral, "Real-time optimal control of an autonomous rc car with minimum-time maneuvers and a novel kineto-dynamical model," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2390–2396.

[25] 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*, vol. 40, pp. 3756–3777, 2024.

[26] 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," *arXiv preprint arXiv:2409.03005*, 2024.

[27] T. Han, A. Liu, A. Li, A. Spitzer, G. Shi, and B. Boots, "Model predictive control for aggressive driving over uneven terrain," *arXiv preprint arXiv:2311.12284*, 2023.

[28] S. Talia, M. Schmittle, A. Lambert, A. Spitzer, C. Mavrogiannis, and S. S. Srinivasa, "Demonstrating hound: A low-cost research platform for high-speed off-road underactuated nonholonomic driving," 2024. [Online]. Available: https://arxiv.org/abs/2311.11199

[29] 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.

[30] A. Dixit, D. D. Fan, K. Otsu, S. Dey, A.-A. Agha-Mohammadi, and J. W. Burdick, "Step: Stochastic traversability evaluation and planning for risk-aware off-road navigation; results from the darpa subterranean challenge," *arXiv preprint arXiv:2303.01614*, 2023.

[31] A. Pokhrel, A. Datar, M. Nazeri, and X. Xiao, "CAHSOR: Competence-aware high-speed off-road ground navigation in SE (3)," *arXiv preprint arXiv:2402.07065*, 2024.

[32] 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.

[33] R. Agishev, K. Zimmermann, V. Kubelka, M. Pecka, and T. Svoboda, "Monoforce: Self-supervised learning of physics-informed model for predicting robot-terrain interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS*, 2024. [Online]. Available: https://arxiv.org/abs/2309.09007

[34] R. Agishev, V. Kubelka, M. Pecka, T. Svoboda, and K. Zimmermann, "End-to-end differentiable model of robot-terrain interactions," in *ICML 2024 Workshop on Differentiable Almost Everything: Differentiable Relaxations, Algorithms, Operators, and Simulators*, 2024. [Online]. Available: https://openreview.net/forum?id=XuVysF8Aon

[35] S. Siva, M. Wigness, J. Rogers, and H. Zhang, "Enhancing consistent ground maneuverability by robot adaptation to complex off-road terrains," in *5th Annual Conference on Robot Learning*, 2021. [Online]. Available: https://openreview.net/forum?id=WIE9t_UwOpM

[36] 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.

[37] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, "Where should i walk? predicting terrain properties from images via self-supervised learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1509–1516, 2019.

[38] A. J. Sathyamoorthy, K. Weerakoon, T. Guan, J. Liang, and D. Manocha, "Terrapn: Unstructured terrain navigation using online self-supervised learning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7197–7204.

[39] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter, "Fast Traversability Estimation for Wild Visual Navigation," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

[40] 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.

[41] A. Zhang, R. Heijne, and J. Biswas, "Lift, splat, map: Lifting foundation masks for label-free semantic scene completion," *arXiv preprint arXiv:2407.03425*, 2024.

[42] H. Karnan, E. Yang, G. Warnell, J. Biswas, and P. Stone, "Wait, that feels familiar: Learning to extrapolate human preferences for

preference-aligned path planning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 13 008–13 014.

[43] R. Schmid, D. Atha, F. Schöller, S. Dey, S. Fakoorian, K. Otsu, B. Ridge, M. Bjelonic, L. Wellhausen, M. Hutter *et al.*, "Self-supervised traversability prediction by learning to reconstruct safe terrain," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 12 419–12 425.

[44] J. Seo, T. Kim, K. Kwak, J. Min, and I. Shim, "Scate: A scalable framework for self-supervised traversability estimation in unstructured environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 888–895, 2023.

[45] G. Kahn, P. Abbeel, and S. Levine, "Badgr: An autonomous self-supervised learning-based navigation system," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1312–1319, 2021.

[46] 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.

[47] H. Karnan, E. Yang, G. Warnell, J. Biswas, and P. Stone, "Wait, that feels familiar: Learning to extrapolate human preferences for preference-aligned path planning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 13 008–13 014.

[48] H. Karnan, E. Yang, D. Farkash, G. Warnell, J. Biswas, and P. Stone, "Sterling: Self-supervised terrain representation learning from unconstrained robot experience," in *Conference on Robot Learning*. PMLR, 2023, pp. 2393–2413.

[49] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[50] J. Frey, M. Mattamala, N. Chebrolu, C. Cadena, M. Fallon, and M. Hutter, "Fast traversability estimation for wild visual navigation," *arXiv preprint arXiv:2305.08510*, 2023.

[51] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[52] A. Payandeh, K. T. Baghaei, P. Fayyazsanavi, S. B. Ramezani, Z. Chen, and S. Rahimi, "Deep representation learning: Fundamentals, technologies, applications, and open challenges," *IEEE Access*, vol. 11, pp. 137 621–137 659, 2023.

[53] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine, "Octo: An open-source generalist robot policy," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[54] R. Doshi, H. Walke, O. Mees, S. Dasari, and S. Levine, "Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation," in *Conference on Robot Learning*, 2024.

[55] A. Radford, "Improving language understanding by generative pre-training," 2018.

[56] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[57] S. Kolouri, P. E. Pope, C. E. Martin, and G. K. Rohde, "Sliced wasserstein auto-encoders," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=H1xaJn05FQ

[58] M. H. Nazeri and M. Bohlouli, "Exploring reflective limitation of behavior cloning in autonomous vehicles," in *2021 IEEE International Conference on Data Mining (ICDM)*, 2021, pp. 1252–1257.

[59] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, Jun. 2019.

[60] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," Jan. 2019.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

[62] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter, "Elevation mapping for locomotion and navigation using gpu," 2022.

[63] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 619–15 629.