VERTIFORMER: A Data-Efficient Multi-Task Transformer on Vertically Challenging Terrain

Mohammad Nazeri[1], **Anuj Pokhrel**[1], **Alexandyr Card**[1], **Aniket Datar**[1], **Garrett Warnell**[2,3], **and Xuesu Xiao**[1]
[1]George Mason University, [2]DEVCOM Army Research Laboratory,
[3]The University of Texas at Austin

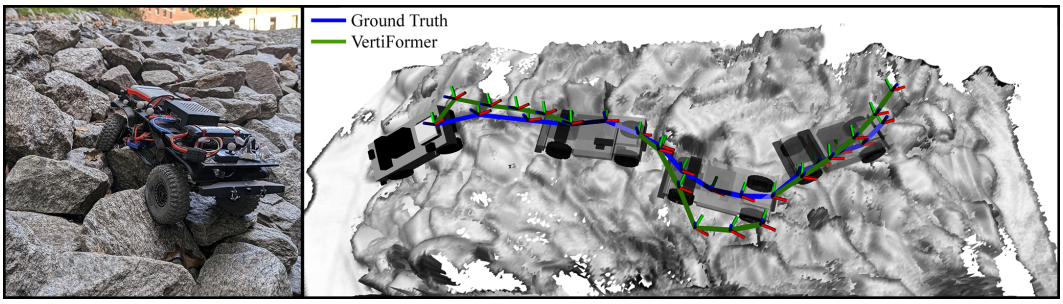


Figure 1: VERTIFORMER is a data-efficient multi-task Transformer for off-road mobility on vertically challenging terrain. VERTIFORMER employs unified multi-modal latent representation, missing modality infilling, and non-autoregressive training to learn complex and nuanced vehicle-terrain interactions in $\mathbb{SE}(3)$ with only one hour of training data.

Abstract: We propose VERTIFORMER, a novel data-efficient multi-task Transformer trained with only one hour of multi-modal data to address the challenges of applying Transformers for robot mobility on extremely rugged, vertically challenging, off-road terrain. With a Transformer encoder and decoder to predict the next robot pose, action, and terrain patch, VERTIFORMER employs a unified state space and missing modality infilling to respectively enhance dynamics understanding and enable a variety of off-road mobility tasks simultaneously, e.g., forward and inverse kinodynamics modeling. By leveraging this unified representation alongside modality infilling, it also achieves real-time task switching during inference for improved fault tolerance and better generalization to unseen environments. Furthermore, VERTIFORMER's non-autoregressive design also mitigates computational bottlenecks and error propagation associated with autoregressive models. Our experiments offer insights into effectively utilizing Transformers for off-road robot mobility with limited data and demonstrate VERTIFORMER can facilitate multiple off-road mobility tasks onboard a physical mobile robot.[1]

Keywords: Navigation, Off-Road Mobility, Representation Learning

1 Introduction

Autonomous mobile robots deployed in off-road environments face significant challenges posed by the underlying terrain. For example, irregular terrain topographies featuring vertical protrusions from the ground, i.e., vertically challenging terrain, pose extensive mobility risks [1, 2, 3], manifesting in several critical ways: compromised chassis stability, leading to potential rollover; increased wheel slippage, resulting in reduced traction and impaired locomotion; and unpredictable vehicle immobilization, causing the robot get stuck, when interacting with vertically challenging terrain.

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

Precisely understanding the vehicle-terrain kinodynamic interactions is the key to mitigating such mobility challenges posed by off-road, vertically challenging terrain. Although data-driven approaches have shown promises in enabling off-road mobility in relatively flat environments [4, 5, 6, 7, 8, 9, 10, 1, 11, 12, 13, 14, 15, 16], the intricate relationships between the robot chassis and vertically challenging terrain, e.g., suspension travel, tire deformation, changing normal and friction forces, and vehicle weight distribution and momentum, motivate the adoption of more sophisticated learning architectures to fully capture and represent the nuanced off-road kinodynamics [3].

Transformers are the preferred architectures to understand complex relationships, which show promise in Natural Language Processing (NLP) [17, 18, 19, 20] and Computer Vision (CV) [21, 22, 23, 24, 25, 26] with self-supervised pre-training emerging as a dominant methodology. This trend is now extending to robotics, impacting areas such as manipulation [27, 28, 29, 30, 31] and autonomous driving [32, 33, 34, 35, 36, 37, 38]. In addition to the advent of the well-studied Transformer architecture [39, 40], this progress is largely attributable to the availability of large-scale datasets [27, 41, 42] as well as various Transformer training techniques including two primary pre-training paradigms: (i) Masked Modeling (MM) and (ii) Next-Token Prediction (NTP) [43].

The application of these paradigms to robotics is particularly limited due to the inherent challenge associated with acquiring large-scale robotics datasets, especially for outdoor, off-road environments. The multi-modal nature of robotics data also presents another significant challenge for Transformers to learn inter-modal relationships and understand the temporal progression of both the environment and the robot state at the same time. These two challenges of applying Transformers to robotics lead to our question: "*How can we train Transformers with limited multi-modal robotics data?*"

Motivated by this research question, this work presents VERTIFORMER, a novel data-efficient multi-task Transformer for robot mobility on extremely rugged, vertically challenging, off-road terrain that requires precisely understanding the kinodynamics in $\mathbb{SE}(3)$ to avoid getting stuck or rolling over. VERTIFORMER's novel unified latent representation of robot exteroception, proprioception, and action offers a stronger inductive bias and therefore off-loads the learning of inter-modality relationships from the Transformer. This consequently facilitates more effective learning with only one hour of data, contrasting current data-intensive methods in NLP, CV, and previous work in robotics [44, 45] that employ separate tokenization of modalities and depend solely on self-attention to capture complex inter-modal correlations within massive datasets. Furthermore, VERTIFORMER's missing modality infilling enables various off-road mobility tasks within one model simultaneously without the need to retrain separate downstream tasks and mitigates the impact of missing modalities at inference time. Additionally, the non-autoregressive nature of VERTIFORMER avoids error propagation from earlier to later prediction steps and makes VERTIFORMER faster at inference because it does not require iterative queries for each step.

VERTIFORMER outperforms the navigation performance achieved by state-of-the-art kinodynamic modeling approaches specifically designed for vertically challenging terrain [46, 47] as well as general navigation models such as NoMaD [48], providing empirical evidence supporting the feasibility of training a Transformer on limited robotic datasets using effective training strategies. Our contributions can be summarized as follows:

- VERTIFORMER, a data-efficient, multi-task Transformer for off-road robot mobility on vertically challenging terrain in $\mathbb{SE}(3)$;
- a unified representation approach to treat all modalities as one single distribution to off-load inter-modality relationship learning from the otherwise data-intensive Transformer;
- a missing modality infilling method that facilitates information sharing among multiple heads and therefore enables different off-road mobility tasks, i.e., forward/inverse kinodynamic learning (FKD/IKD) and zero-shot navigation policy (NP);
- an extensive evaluation of different Transformer designs, including MM, NTP, Encoder-only, and Decoder-only, for off-road kinodynamic representation learning; and
- physical on-robot experiments for different off-road mobility tasks on vertically challenging terrain and comparison against state-of-the-art methods.

2 Related Work

Transformers, initially proposed for language translation tasks, have demonstrated remarkable versatility across a spectrum of domains, including CV and robotics. This section provides an overview of existing work on Transformers in robotics and data-driven off-road mobility. We provide more details on the current best practices with Transformers in NLP and CV in Sec. 8 of the Appendix.

Transformers in Robotics. Recent years have witnessed a surge in the application of Transformers to robotics, encompassing both perception and planning: Generalist robot policies based on Transformers, e.g., Octo [49] and CrossFormer [44], with multi-modal sensory input [50] and action tokenization [51] aimed at handling diverse tasks such as manipulation and navigation; Studies in target-driven [52, 53, 54, 55] and image-goal navigation [56, 57] have shown that Transformers significantly outperform traditional behavior cloning baselines [58, 59]; Reinforcement learning has been significantly enhanced by integrating the Transformer architecture, providing improved sequence modeling [60] and decision-making capabilities [61]; Transformers have also been used in motion planning to guide long-horizon navigation tasks [62] and reduce the search space for sampling-based motion planners [63]; In Unmanned Surface Vehicles (USV), MarineFormer [64] utilized Transformers to learn the flow dynamics around a USV and then learned a navigation policy resulting in better path length and completion rate.

A common characteristic of these models is their treatment of each sensor modality (e.g., vision, touch, and audio) as a distinct token, relying on the Transformer to learn the inter-modal correlations and their temporal dynamics. While this approach allows for flexible integration of diverse sensory information, it necessitates substantial amounts of training data to compensate for the lack of inductive bias inherent in Transformers [40]. This data dependency poses a significant challenge, particularly in off-road robot mobility, where real-world, outdoor data acquisition can be expensive and time-consuming. Consequently, there remains a critical need for research focused on refining training methodologies and exploring architectural modifications specifically tailored to address the data scarcity and multi-modality often encountered in robotics.

Learning Off-Road Mobility. While most learning approaches for off-road autonomy focus on perception tasks [10, 65, 66], researchers have recently investigated off-road mobility to account for vehicle stability [67, 2, 68, 15], wheel slippage [69, 70, 13], and terrain traversability [8, 12, 14, 71, 16]. A relevant work by Xiao et al. [36] used Transformers to enable a universal forward kinodynamic model that can drive different ground vehicles. Most of these approaches have adopted specific techniques designed to address one particular off-road mobility task.

Focusing on multi-task kinodynamic representation for off-road mobility on vertically challenging terrain, our novel non-autoregressive VERTIFORMER employs unified modality latent representation and missing modality infilling to predict the next robot pose, action, and terrain patch in order to simultaneously enable a variety of off-road mobility tasks, i.e., FKD, IKD, NP, and terrain patch reconstruction, without a specific training procedure for each.

3 VERTIFORMER

We introduce VERTIFORMER, a data-efficient multi-task Transformer for kinodynamic representation and navigation on complex, vertically challenging, off-road terrain. We propose an efficient training methodology for training VERTIFORMER utilizing limited (one hour) robotics data, including unified multi-modal latent representation, missing modality infilling, and non-autoregressive training to improve data efficiency and enable multi-task learning.

3.1 VERTIFORMER Training

VERTIFORMER consists of both TransformerEncoder (VERTIENCODER) and TransformerDecoder (VERTIDECODER), as illustrated in Fig. 2 left and right, respectively. Consistent with established practices [46, 47], VERTIFORMER receives a multi-modal sequence of actions $\mathbf{a}_{0:T}$, robot

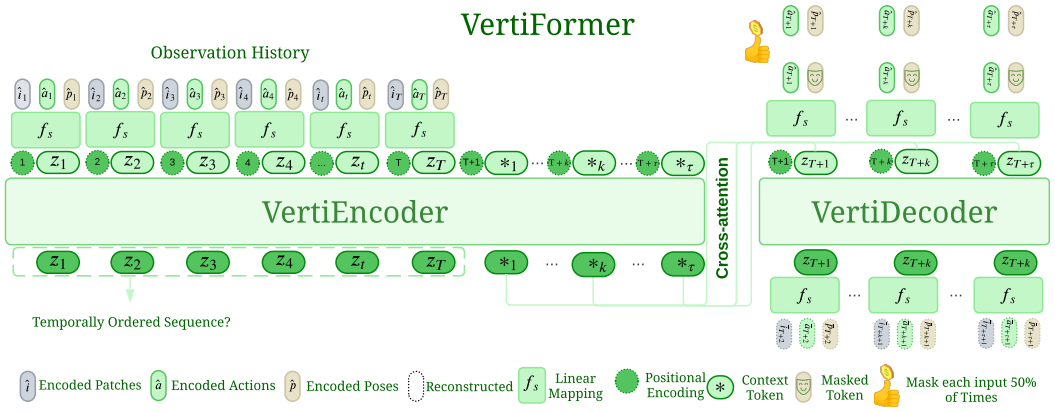


Figure 2: **VERTIFORMER Architecture**. VERTIFORMER employs a TransformerEncoder (left) to receive a history of terrain patches, actions, and poses along with multiple context tokens. To predict future states, the model computes cross-attention between these context tokens and the upcoming actions or poses. VERTIDECODER uses causal masking to ensure that predictions are conditioned only on past and present information, preventing information leakage from future time steps.

poses $\mathbf{p}_{0:T}$, and the underlying terrain patches $\mathbf{i}_{0:T}$. The VERTIENCODER first applies an independent linear mapping to each modality. Specifically, action commands $\mathbf{a}_{0:T}$ are projected into an embedding space via a linear function f_a, yielding $\hat{\mathbf{a}}_{0:T}$. Analogously, robot poses $\mathbf{p}_{0:T}$ and terrain patches $\mathbf{i}_{0:T}$ are transformed using linear mappings f_p and f_i respectively, producing a sequence of embeddings $\hat{\mathbf{p}}_{0:T}$ and $\hat{\mathbf{i}}_{0:T}$. This initial linear mapping can be formally expressed as:

$$\hat{a}_t = f_a(a_t) = W_a a_t + b_a, a_t \in \mathbf{a}_{0:T}, \tag{1}$$
$$\hat{p}_t = f_p(p_t) = W_p p_t + b_p, p_t \in \mathbf{p}_{0:T}, \tag{2}$$
$$\hat{i}_t = f_i(i_t) = W_i i_t + b_i, i_t \in \mathbf{i}_{0:T}, \tag{3}$$

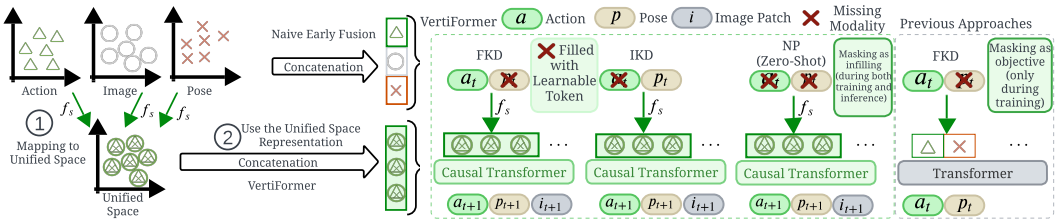
where W_a, W_p, and W_i represent the weight matrices, and b_a, b_p, and b_i denote the bias vectors for each respective modality.

3.1.1 Unified Multi-Modal Latent Representation

To off-load cross-modal interaction learning from Transformer, it is crucial to establish a consistent distributional characteristic across the modality-specific embeddings. Instead of aligning different embeddings, VERTIFORMER treats all modalities as a single unified modality. To achieve this, a subsequent linear transformation, denoted by f_s, is applied to the embeddings:

$$z_t = f_s(\hat{a}_t, \hat{p}_t, \hat{i}_t) = W_s(\hat{a}_t \cdot \hat{p}_t \cdot \hat{i}_t) + b_s, t \in [0:T], \tag{4}$$

with W_s and b_s denoting the weight matrix and bias vector for f_s, respectively. This shared linear mapping f_s aims to project all embeddings into a unified latent space, minimizing potential discrepancies in statistical properties. The resulting unified tokens, $\mathbf{z}_{0:T}$, are then passed as input to the VERTIENCODER (Fig.2 top left). This procedure ensures a homogeneous input representation for the subsequent encoding layers, crucial for effective multi-modal fusion of robotic data (Fig. 3a). This new unified representation stems from the intuition that these input modalities represent the same scene and should therefore share a common representation space. To reinforce this, we also apply tied encoder-decoder weights [72], which further guide the modalities toward a shared distribution. This new modality unification approach results in a more coherent multi-modal representation, leading to improved kinodynamics understanding, particularly in *data-constrained* scenarios. Empirical results (Fig. 4) supporting the importance of such unified representation, in contrast to the conventional individual modality representations, will be presented in Section 4.



(a) Unified State Representation (b) Missing Modality Infilling

Figure 3: The integration of unified state representation and missing modality infilling enables VERTIFORMER to perform simultaneous temporal inference of FKD, IKD, and zero-shot NP.

3.1.2 Missing Modality Infilling for Multi-Task Learning

We propose stochastic modality infilling (Fig. 2, top right) to enable VERTIFORMER's multi-task prediction (i.e., pose, action, navigation, and terrain, Fig. 2, bottom right), aiming for enhanced data efficiency via shared representations.

After a warm-up phase, training involves replacing future (τ steps ahead) poses ($\mathbf{p}_{\mathrm{T+1:T+}\tau}$) or actions ($\mathbf{a}_{\mathrm{T+1:T+}\tau}$) with learnable vectors (50% probability each). This facilitates two tasks: Action-Conditioned Pose Prediction (given actions, predict poses) and Pose-Conditioned Action Prediction (given poses, predict actions), analogous to FKD and IKD respectively.

This strategy promotes a joint action-pose representation as the learnable tokens, processed by f_s and thus aligned with the modality distributions. Consequently, the model supports dynamic task adaptation at inference and infers missing modalities through time (Fig. 3b).

Furthermore, by extending this infilling strategy to replace both future actions, $\mathbf{a}_{\mathrm{T+1:T+}\tau}$, and future poses, $\mathbf{p}_{\mathrm{T+1:T+}\tau}$, simultaneously, VERTIFORMER becomes a navigation policy in a zero-shot manner. In this configuration, the model predicts both actions and poses solely based on the historical context, effectively mimicking the demonstrated behavior without requiring explicit information about future actions and poses from a planner. Notice that compared to masked modeling approaches [73, 47] these learnable vectors are not masked as a learning objective, i.e., masked token reconstruction, instead they act as the modality representation and are present also during inference.

3.1.3 Non-Autoregressive Training

Building upon the work by Octo Model Team et al. [49] and Doshi et al. [44], VERTIFORMER employs multiple context tokens to represent a distribution of plausible future states. These context tokens serve to inform VERTIDECODER in predicting both the future ego state and the evolution of the environment. Having multiple context tokens allows VERTIFORMER to predict the future non-autoregressively. The non-autoregressive nature of VERTIFORMER is motivated by the potential computational bottlenecks inherent in autoregressive models, which require querying the model multiple times and are subject to drifting due to error propagation from earlier steps. By learning multi-context representations, the non-autoregressive approach aims to improve both training efficiency and inference speed—a critical consideration for real-time robotic control applications.

We train VERTIFORMER by minimizing the Mean Squared Error between the model's predictions and the corresponding ground truth values. We evaluate the model by calculating the error rate between the model's predictions and the ground truth values on a held-out, unseen dataset.

3.2 VERTIFORMER Inference

During FKD inference, VERTIENCODER receives the same historical input as training. VERTIDECODER receives sampled actions from an external sampling-based planner (e.g., MPPI [74]) while masking the corresponding poses, compelling the model to predict future poses based solely on the sampled actions (and the context tokens) so that the planner can choose the optimal trajectory to minimize a cost function. For IKD, a global planner generates desired future poses, and by mask-

ing the actions we encourage the model to predict future actions to achieve these globally planned poses. By masking both actions and poses, VERTIFORMER can still navigate by predicting actions in a zero-shot NP manner. We provide VERTIFORMER's architecture parameters in Appendix 9, and implementation details along with the one-hour dataset description are provided in Appendix 10. Qualitative samples of FKD are provided in Fig. 10 of Appendix 11.

4 Training on One Hour of Data

VERTIFORMER's one hour of training data comes from a human-teleoperated demonstration of driving an open-source four-wheeled ground vehicle [3], Verti-4-Wheeler (V4W), on a custom-built off-road testbed composed of hundreds of rocks and boulders. The demonstrator mostly aims to drive the robot to safely and stably traverse the vertically challenging terrain, but still occasionally encounters dangerous situations such as large roll angles and getting stuck between rocks. Fortunately, those situations serve as explorations for VERTIFORMER to understand a wider range of kinodynamic interactions. Direct application of standard Transformer training methodologies in NLP and CV to such a small robotics dataset proves challenging due to the inherent lack of inductive bias in Transformers [40], which necessitates substantial amounts of data for effective training. However, our experiments suggest that VERTIFORMER's judicious modifications to established MM and NTP training paradigms can facilitate effective Transformer training even with limited robotics data.

Unified latent space representation facilitates FKD, IKD, and NP by decoupling inter-modality learning from temporal progression modeling, with only the latter handled by the Transformer, which otherwise becomes data-intensive.

To evaluate VERTIFORMER's dynamics understanding, we use a sequence order prediction task where the model classifies sequences as chronologically ordered (50%) or randomly shuffled (50%). This probes the model's grasp of temporal dependencies and kinodynamics evolution.

As illustrated (Fig. 4), non-unified tokens result in poor kinodynamic understanding (minimal loss decrease) and suggest that fragmented processing hinders capturing temporal relationships. While larger datasets might compensate, they are often unavailable in robotics.

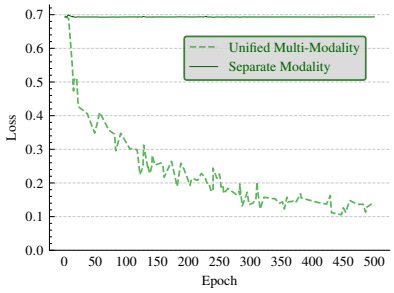


Figure 4: Without unified latent representation the model cannot capture temporal dependencies and understand kinodynamic transitions, resulting in an almost flat learning curve.

Conversely, the unified representation significantly improves the model's ability to discern temporal order and understand system dynamics by consolidating information cohesively. This underscores the importance of unified representations for learning complex dynamics effectively from limited robotics data, unlike in data-rich NLP/CV domains.

Longer prediction horizons in navigation planning improve foresight but increase uncertainty via error accumulation, especially in autoregressive models like VERTIDECODER where errors propagate. We compare the autoregressive VERTIDECODER with the non-autoregressive VERTIFORMER on long-horizon accuracy. Results (Fig. 5) show VERTIFORMER predicts longer (2s) with less drift than VERTIDECODER predicting shorter (1s), highlighting the advantage of non-autoregressive models for reducing compounding errors in long-term predictions. We provide qualitative results as well in Fig. 11 of the Appendix.

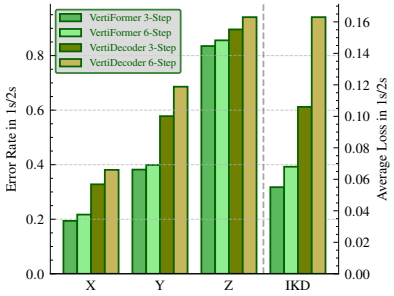


Figure 5: VERTIFORMER is capable of predicting a longer horizon without losing much accuracy due to its non-autoregressive nature.

MM vs NTP vs End2End are currently the prominent approaches in CV, NLP, and robotics respectively. We compare MM, NTP, and End2End for off-road mobility tasks. We analyze an MM encoder

Table 1: Physical results with VERTIFORMER, VERTICODER, VERTIDECODER, NoMaD, and TAL .

Task	Model	**SR** ↑	**TT** ↓	**Roll** ↓	**Pitch** ↓
FKD	TAL [46]	8/10	11.80 ± 0.87	0.198 ± 0.38	$\mathbf{0.086} \pm \mathbf{0.07}$
	VERTIDECODER	6/10	15.12 ± 1.78	0.180 ± 0.30	0.114 ± 0.09
	VERTICODER [47]	**10**/10	$\mathbf{8.58} \pm 1.54$	0.189 ± 0.23	0.116 ± 0.08
	VERTIFORMER	**10**/10	$9.42 \pm \mathbf{0.61}$	$\mathbf{0.169} \pm \mathbf{0.17}$	0.096 ± 0.08
IKD	VERTIDECODER	**10**/10	$15.92 \pm \mathbf{1.08}$	0.181 ± 0.23	0.125 ± 0.08
	VERTICODER [47]	7/10	$\mathbf{13.99} \pm 3.27$	$\mathbf{0.136} \pm 0.14$	$\mathbf{0.069} \pm \mathbf{0.07}$
	VERTIFORMER	8/10	17.16 ± 6.10	$\mathbf{0.136} \pm \mathbf{0.10}$	$0.077 \pm \mathbf{0.07}$
NP	NoMaD [48]	1/10	22.3	0.187	0.09
	NoMaD-*scratch*	0/10	-	-	-
	VERTICODER [47]	**9**/10	$13.49 \pm \mathbf{3.33}$	0.175 ± 0.37	$\mathbf{0.089} \pm 0.09$
	VERTIFORMER	8/10	$\mathbf{12.64} \pm 3.89$	$\mathbf{0.154} \pm \mathbf{0.11}$	$0.099 \pm \mathbf{0.08}$

(VERTICODER [47]), an autoregressive NTP decoder (VERTIDECODER, Fig. 2 right trained alone without cross-attention), a non-Transformer End2End model [47], and VERTIFORMER, our non-autoregressive Transformer (Fig. 2). VERTICODER and VERTIDECODER use the unified representation (Fig. 4). The End2End model employs ResNet-18 [75] for computational balance (Appendix 9).

Evaluations (Fig. 6, 1s horizon) show VERTIFORMER achieves superior performance on FKD, IKD, and NP error rates. Its non-autoregressive prediction leads to better accuracy than the autoregressive VERTIDECODER (which cannot directly perform NP, as it has access to both action and pose at each step). VERTIFORMER's joint multi-task training also surpasses VERTICODER's separate training [47] (except Z prediction). The End2End model exhibits the highest errors, highlighting Transformer's benefits for kinodynamics learning.

Beyond accuracy, VERTIFORMER supports concurrent multi-task execution during inference, vital for real-time robotics, especially with missing modalities (e.g., sensor degradation and planner failure). We provide more experiments on the analysis of basic factors to train Transformers in general in Appendix 8.

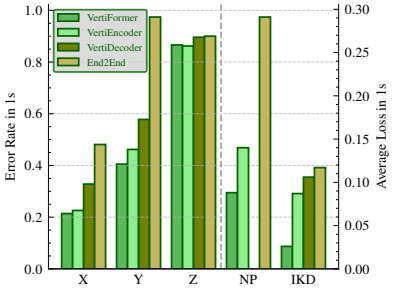


Figure 6: VERTIFORMER achieves the best accuracy across FKD, IKD, and NP compared to VERTICODER (MM), VERTIDECODER (NTP), and End2End.

5 Robot Experiments

We implement VERTIFORMER's FKD, IKD, and NP on the V4W ground robot platform. The experiments are carried out on a *never* before seen 4 m $\times$ 2.5 m testbed made of rocks/boulders, wooden planks, AstroTurf with crumpled cardboard boxes underneath, and modular 0.8 m $\times$ 0.75 m expanding foam to represent different types of vertically challenging terrain with different friction coefficients and varying deformability (Fig. 7 of Appendix). The modular foam and rocks/boulders do not deform, while the rocks may shift positions under the weight of the robot. On the other hand, the wooden planks and AstroTurf are completely deformable and change the terrain topography during wheel-terrain interactions. The one-hour training dataset used (see details of the dataset in Appendix 10) only consists of robot teleoperation on the rigid rock/boulder testbed and hence the experiment testbed is an unseen environment, posing generalization challenges for VERTIFORMER. Details of FKD, IKD, and NP implementations are provided in Appendix 10.2.

Results and Discussions. The results of the three methods are then compared to MPPI using TAL [46], a highly accurate forward kinodynamic model specifically designed for vertically challenging terrain, and NoMaD [48], a state-of-the-art general navigation model based on diffusion policy. We train NoMaD from scratch (NoMaD-*scratch* in Table 1) to illustrate the difficulty of learning from our limited (one hour) data, while comparison with pre-trained NoMaD highlights

the inadequacy of 2D assumptions for vertically challenging terrain, necessitating an understanding of 3D robot-terrain interactions. Since NoMaD tackles a different problem than VERTIFORMER, fine-tuning its pre-trained weights with data from an unrelated task would negatively impact its performance. We report success rate (SR), traversal time (TT), and roll and pitch angles in Table 1.

Our observations reveal a nuanced performance difference between VERTICODER [47] and VERTIFORMER, particularly concerning NP and IKD. VERTICODER leverages MM pre-training to learn a general kinodynamic representation. Then it trains separate downstream task heads with the learned representation, providing VERTICODER with privileged information for each task. This specialized training allows VERTICODER to effectively leverage the provided data for NP. In contrast, VERTIFORMER approaches NP in a zero-shot manner. It is not explicitly trained on NP, relying instead on its modality infilling strategy. This infilling effectively handles missing modalities by replacing them with a trained mask, enabling the model to infer behavior without direct NP training. While this approach allows VERTIFORMER to perform NP without specialized training, it also explains why VERTICODER, with its dedicated head, achieves a higher success rate. A similar trend is observed with IKD. VERTIDECODER has access to both predicted and actual actions and poses at each time step, providing richer guidance for the IKD process. This richer information stream in VERTIDECODER is the reason for achieving a higher success rate, especially considering the inherent difficulty of IKD compared to FKD. VERTIFORMER, however, faces a challenge in IKD and takes longer to finish the traversal. The infilling strategy, while effective for missing modality, is not as accurate as the actual modality.

Regarding FKD, the architectural difference between VERTIFORMER and VERTICODER causes different navigation behaviors. VERTICODER's specialized task head for FKD treats each future step independently without any attention weights between steps. While this approach facilitates faster MPPI initial convergence due to a lack of cross attention, it can also lead to drift, causing inconsistencies between predicted steps and ultimately resulting in a larger standard deviation of traversal time across trials. While VERTICODER's MPPI converges quickly, it struggles with long-term consistency. VERTIFORMER takes a different approach. By employing attention and cross-attention mechanisms between historical and future steps, it dynamically incorporates past information into future predictions. This allows VERTIFORMER to consider the historical context through cross-attention and causal masking when predicting future states, leading to more coherent and consistent predictions. Consequently, although MPPI might require more time to converge on a path with VERTIFORMER, once it does, the resulting behavior is more robust and less variable across trials, reflected in a smaller traversal time standard deviation.

6 Conclusions

In this work, we introduce VERTIFORMER, a novel data-efficient multi-task Transformer designed for learning kinodynamic representations on vertically challenging, off-road terrain. VERTIFORMER demonstrates the capacity to simultaneously address forward kinodynamics learning, inverse kinodynamics learning, and navigation policy learning tasks, only using one hour of training data. Key contributions include a unified latent space representation enhancing temporal understanding, learned modality infilling facilitating multiple off-road mobility tasks simultaneously and acting as a proxy for missing modalities during inference, and multi-context tokens enabling multi-step prediction without autoregressive feedback. All three contributions improve robustness and generalization of VERTIFORMER to out-of-distribution environments. We provide extensive experiment results and empirical guidelines for training Transformers under extreme data scarcity. Our evaluations across all three downstream tasks demonstrate that VERTIFORMER outperforms baseline models, including TAL [46], VERTICODER [47], VERTIDECODER, NoMaD [48], and end-to-end approaches, while exhibiting reduced overfitting and improved generalization and highlighting the efficacy of the proposed architecture and training methodology for learning kinodynamic representations in data-constrained settings. Physical experiments also demonstrate that VERTIFORMER can enable superior off-road robot mobility on vertically challenging terrain. We leave extending this work to general navigation as future work.

7 Limitations

VERTIFORMER can capture long-range dependencies through additional context tokens, but it requires re-training if we want to change the prediction horizon, while autoregressive models can predict any number of steps into the future without re-training. However, it is possible to treat VERTIFORMER as an autoregressive model during inference and predict longer horizons without the need for re-training. As illustrated in Fig. 10 of Appendix 11, our model demonstrates a deficiency in accurately executing a turning maneuver. Such failures stem from long-horizon (1 second), non-autoregressive predictions in one step accentuated by the inaccuracy of terrain reconstruction caused by the high degree of complexity present in off-road topographical formations. This also reflects on the accuracy of predicting vehicle $\mathbf{Z}$. On 2D surfaces, this should not pose a problem. A further limitation comes from the unified state representation where adding new modalities requires training the model from scratch.

It is crucial to acknowledge that our observations are primarily associated with the challenges inherent in wheeled locomotion on complex, vertically challenging, off-road terrain that requires an understanding of the robot-terrain interactions in 3D and may not be applicable to other robotic domains such as visual navigation or manipulation without further investigation. In visual navigation, the robot typically relies on visual cues and image processing to perceive its environment and plan its path. In manipulation tasks, the focus is on interacting with objects rather than negotiating through complex terrain. Further investigation is required for general visual navigation and manipulation.

References

[1] P. Borges, T. Peynot, S. Liang, B. Arain, M. Wildie, M. Minareci, S. Lichman, G. Samvedi, I. Sa, N. Hudson, M. Milford, P. Moghadam, and P. Corke. A survey on terrain traversability analysis for autonomous ground vehicles: Methods, sensors, and challenges. *Field Robotics*, 2:1567–1627, 2022. doi:10.55417/fr.2022049.

[2] 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.

[3] 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)*, pages 16322–16329, May 2024. doi:10.1109/ICRA57147.2024.10610079.

[4] T. Overbye and S. Saripalli. Fast local planning and mapping in unknown off-road terrain. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5912–5918. IEEE, 2020.

[5] 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*, 2020.

[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*, 6(3):6054–6060, 2021.

[7] 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)*, pages 4844–4850. IEEE, 2021.

[8] D. D. Fan, K. Otsu, Y. Kubo, A. Dixit, J. Burdick, and A.-A. Agha-Mohammadi. Step: Stochastic traversability evaluation and planning for risk-aware off-road navigation. In *Robotics: Science and Systems (RSS)*, 2021.

[9] 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)*, pages 3294–3301. IEEE, 2022.

[10] X. Xiao, B. Liu, G. Warnell, and P. Stone. Motion planning and control for mobile robot navigation using machine learning: A survey. *Autonomous Robots*, 46(5):569–597, June 2022. ISSN 0929-5593, 1573-7527. doi:10.1007/s10514-022-10039-8.

[11] 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)*, pages 4452–4458. IEEE, 2022.

[12] S. Triest, M. Sivaprakasam, S. J. Wang, W. Wang, A. M. Johnson, and S. Scherer. Tartandrive: A large-scale dataset for learning off-road dynamics models. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2546–2552. IEEE, 2022.

[13] L. Sharma, M. Everett, D. Lee, X. Cai, P. Osteen, and J. P. How. Ramp: A risk-aware mapping and planning pipeline for fast off-road ground robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5730–5736. IEEE, 2023.

[14] 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)*, pages 931–938. IEEE, 2023.

[15] 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*, 9(11):9653–9660, 2024.

[16] 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.

[17] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. 2018.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi:10.18653/v1/N19-1423.

[19] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

[20] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[21] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[22] C. Feichtenhofer, H. Fan, Y. Li, and K. He. Masked Autoencoders As Spatiotemporal Learners, May 2022.

[23] X. Geng, H. Liu, L. Lee, D. Schuurmans, S. Levine, and P. Abbeel. Multimodal Masked Autoencoders Learn Transferable Representations, May 2022.

[24] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. DINOv2: Learning Robust Visual Features without Supervision, Apr. 2023.

[25] E. Karypidis, I. Kakogeorgiou, S. Gidaris, and N. Komodakis. DINO-Foresight Looking into the Future with DINO, Dec. 2024.

[26] V. Pătrăucean, X. O. He, J. Heyward, C. Zhang, M. S. M. Sajjadi, G.-C. Muraru, A. Zholus, M. Karami, R. Goroshin, Y. Chen, S. Osindero, J. Carreira, and R. Pascanu. TRecViT: A Recurrent Video Transformer, Dec. 2024.

[27] 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. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[28] Y. Du, M. Yang, P. Florence, F. Xia, A. Wahid, B. Ichter, P. Sermanet, T. Yu, P. Abbeel, J. B. Tenenbaum, L. Kaelbling, A. Zeng, and J. Tompson. Video Language Planning, Oct. 2023.

[29] Y. Seo, D. Hafner, H. Liu, F. Liu, S. James, K. Lee, and P. Abbeel. Masked World Models for Visual Control, May 2023.

[30] Y. Seo, J. Kim, S. James, K. Lee, J. Shin, and P. Abbeel. Multi-View Masked World Models for Visual Robotic Manipulation, Feb. 2023.

[31] Y. Hu, Y. Guo, P. Wang, X. Chen, Y.-J. Wang, J. Zhang, K. Sreenath, C. Lu, and J. Chen. Video Prediction Policy: A Generalist Robot Policy with Predictive Visual Representations, Dec. 2024.

[32] A. Hu, L. Russell, H. Yeo, Z. Murez, G. Fedoseev, A. Kendall, J. Shotton, and G. Corrado. GAIA-1: A Generative World Model for Autonomous Driving, Sept. 2023.

[33] J. Mao, Y. Qian, H. Zhao, and Y. Wang. GPT-Driver: Learning to Drive with GPT, Oct. 2023.

[34] X. Hu, W. Yin, M. Jia, J. Deng, X. Guo, Q. Zhang, X. Long, and P. Tan. DrivingWorld: Constructing World Model for Autonomous Driving via Video GPT, Dec. 2024.

[35] A. Bar, G. Zhou, D. Tran, T. Darrell, and Y. LeCun. Navigation World Models, Dec. 2024.

[36] W. Xiao, H. Xue, T. Tao, D. Kalaria, J. M. Dolan, and G. Shi. AnyCar to Anywhere: Learning Universal Dynamics Model for Agile and Adaptive Mobility, Sept. 2024.

[37] M. Mattamala, J. Frey, P. Libera, N. Chebrolu, G. Martius, C. Cadena, M. Hutter, and M. Fallon. Wild Visual Navigation: Fast Traversability Learning via Pre-Trained Models and Online Self-Supervision, Apr. 2024.

[38] B. Ai, Z. Wu, and D. Hsu. Invariance is Key to Generalization: Examining the Role of Representation in Sim-to-Real Transfer for Visual Navigation, Dec. 2023.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[40] 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, June 2021.

[41] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[42] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[43] L. Chen, Z. Wang, S. Ren, L. Li, H. Zhao, Y. Li, Z. Cai, H. Guo, L. Zhang, Y. Xiong, Y. Zhang, R. Wu, Q. Dong, G. Zhang, J. Yang, L. Meng, S. Hu, Y. Chen, J. Lin, S. Bai, A. Vlachos, X. Tan, M. Zhang, W. Xiao, A. Yee, T. Liu, and B. Chang. Next Token Prediction Towards Multimodal Intelligence: A Comprehensive Survey, Dec. 2024.

[44] R. Doshi, H. Walke, O. Mees, S. Dasari, and S. Levine. Scaling Cross-Embodied Learning: One Policy for Manipulation, Navigation, Locomotion and Aviation, Aug. 2024.

[45] J. Jones, O. Mees, C. Sferrazza, K. Stachowicz, P. Abbeel, and S. Levine. Beyond sight: Finetuning generalist robot policies with heterogeneous sensors via language grounding. *arXiv preprint arXiv:2501.04693*, 2025.

[46] 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)*, pages 5438–5443, Abu Dhabi, United Arab Emirates, Oct. 2024. IEEE. ISBN 979-8-3503-7770-5. doi:10.1109/IROS58592.2024.10801650.

[47] M. Nazeri, A. Datar, A. Pokhrel, C. Pan, G. Warnell, and X. Xiao. VertiCoder: Self-Supervised Kinodynamic Representation Learning on Vertically Challenging Terrain, Sept. 2024.

[48] A. Sridhar, D. Shah, C. Glossop, and S. Levine. NoMaD: Goal Masked Diffusion Policies for Navigation and Exploration, Oct. 2023.

[49] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, J. Luo, Y. L. Tan, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An Open-Source Generalist Robot Policy, May 2024.

[50] J. Jones, O. Mees, C. Sferrazza, K. Stachowicz, P. Abbeel, and S. Levine. Beyond Sight: Finetuning Generalist Robot Policies with Heterogeneous Sensors via Language Grounding, Jan. 2025.

[51] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models, Jan. 2025.

[52] H. Du, X. Yu, and L. Zheng. VTNet: Visual Transformer Network for Object Goal Navigation, May 2021.

[53] H. Wang, A. H. Tan, and G. Nejat. NavFormer: A Transformer Architecture for Robot Target-Driven Navigation in Unknown and Dynamic Environments. 2024. doi:10.48550/ARXIV.2402.06838.

[54] 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)*, pages 2741–2746, Abu Dhabi, United Arab Emirates, Oct. 2024. IEEE. ISBN 979-8-3503-7770-5. doi:10.1109/IROS58592.2024.10802451.

[55] W. Huang, Y. Zhou, X. He, and C. Lv. Goal-Guided Transformer-Enabled Reinforcement Learning for Efficient Autonomous Navigation. *IEEE Transactions on Intelligent Transportation Systems*, 25(2):1832–1845, Feb. 2024. ISSN 1558-0016. doi:10.1109/TITS.2023.3312453.

[56] N. Pelluri. Transformers for Image-Goal Navigation, May 2024.

[57] X. Liu, J. Li, Y. Jiang, N. Sujay, Z. Yang, J. Zhang, J. Abanes, J. Zhang, and C. Feng. CityWalker: Learning Embodied Urban Navigation from Web-Scale Videos, Nov. 2024.

[58] D. A. Pomerleau. ALVINN: An autonomous land vehicle in a neural network. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 1. Morgan-Kaufmann, 1988.

[59] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba. End to End Learning for Self-Driving Cars, Apr. 2016.

[60] X. Zhang, Z. Feng, Q. Qiu, Y. Chen, B. Hua, and J. Ji. NaviFormer: A Data-Driven Robot Navigation Approach via Sequence Modeling and Path Planning with Safety Verification. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14756–14762, May 2024. doi:10.1109/ICRA57147.2024.10610076.

[61] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch. Decision Transformer: Reinforcement Learning via Sequence Modeling. *arXiv:2106.01345 [cs]*, June 2021.

[62] D. Lawson and A. H. Qureshi. Control Transformer: Robot Navigation in Unknown Environments Through PRM-Guided Return-Conditioned Sequence Modeling. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9324–9331, Oct. 2023. doi:10.1109/IROS55552.2023.10341628.

[63] J. J. Johnson, U. S. Kalra, A. Bhatia, L. Li, A. H. Qureshi, and M. C. Yip. Motion Planning Transformers: A Motion Planning Framework for Mobile Robots, Nov. 2022.

[64] E. Kazemi and I. Soltani. MarineFormer: A Spatio-Temporal Attention Model for USV Navigation in Dynamic Marine Environments, Dec. 2024.

[65] M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon. A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5000–5007. IEEE, 2019.

[66] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli. Rellis-3d dataset: Data, benchmarks and analysis. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 1110–1116. IEEE, 2021.

[67] J. Bae, T. Kim, W. Lee, and I. Shim. Curriculum learning for vehicle lateral stability estimations. *IEEE Access*, 9:89249–89262, 2021.

[68] A. Datar, C. Pan, and X. Xiao. Learning to model and plan for wheeled mobility on vertically challenging terrain. *IEEE Robotics and Automation Letters*, 10(2):1505–1512, 2025.

[69] S. Siva, M. Wigness, J. Rogers, and H. Zhang. Robot adaptation to unstructured terrains by joint representation and apprenticeship learning. In *Robotics: Science and Systems (RSS)*, 2019.

[70] S. Siva, M. Wigness, J. G. Rogers, L. Quang, and H. Zhang. Nauts: Negotiation for adaptation to unstructured terrain surfaces. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1733–1740. IEEE, 2022.

[71] J. Seo, S. Sim, and I. Shim. Learning Off-Road Terrain Traversability with Self-Supervisions Only, May 2023.

[72] O. Press and L. Wolf. Using the output embedding to improve language models. In M. Lapata, P. Blunsom, and A. Koller, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain, Apr. 2017. Association for Computational Linguistics.

[73] J. Ngiam, V. Vasudevan, B. Caine, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, D. J. Weiss, B. Sapp, Z. Chen, and J. Shlens. Scene transformer: A unified architecture for predicting future trajectories of multiple agents. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Wm3EA5OlHsG.

[74] G. Williams, A. Aldrich, and E. A. Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 2017.

[75] 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)*, pages 770–778, 2015. doi:10.1109/cvpr.2016.90.

[76] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020. doi:10.5555/3524938.3525913.

[77] P. Xu, D. Kumar, W. Yang, W. Zi, K. Tang, C. Huang, J. C. K. Cheung, S. J. Prince, and Y. Cao. Optimizing Deeper Transformers on Small Datasets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2089–2102, Online, 2021. Association for Computational Linguistics. doi:10.18653/v1/2021.acl-long.163.

[78] I. Loshchilov, C.-P. Hsieh, S. Sun, and B. Ginsburg. nGPT: Normalized Transformer with Representation Learning on the Hypersphere, Oct. 2024.

[79] X. Chen, S. Xie, and K. He. An Empirical Study of Training Self-Supervised Vision Transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9620–9629, Montreal, QC, Canada, Oct. 2021. IEEE. ISBN 978-1-6654-2812-5. doi:10.1109/ICCV48922.2021.00950.

[80] Y. Liu, E. Sangineto, W. Bi, N. Sebe, B. Lepri, and M. Nadai. Efficient training of visual transformers with small datasets. *Advances in Neural Information Processing Systems*, 34:23818–23830, 2021.

[81] H. Gani, M. Naseer, and M. Yaqub. How to train vision transformer on small-scale datasets? In *33rd British Machine Vision Conference Proceedings, BMVC 2022*, 2022.

[82] A. P. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer. How to train your ViT? Data, augmentation, and regularization in vision transformers. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

[83] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever. Generative pretraining from pixels. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 2020-07-13/2020-07-18.

[84] C. Mao, L. Jiang, M. Dehghani, C. Vondrick, R. Sukthankar, and I. Essa. Discrete Representations Strengthen Vision Transformer Robustness, Apr. 2022.

[85] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12104–12113, June 2022.

[86] 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. *arXiv preprint arXiv:2301.08243*, 2023.

[87] A. Bardes, J. Ponce, and Y. LeCun. MC-JEPA: A Joint-Embedding Predictive Architecture for Self-Supervised Learning of Motion and Content Features, July 2023.

[88] A. Bardes, Q. Garrido, J. Ponce, X. Chen, M. Rabbat, Y. LeCun, M. Assran, and N. Ballas. Revisiting Feature Prediction for Learning Visual Representations from Video, Feb. 2024.

[89] J. Rajasegaran, I. Radosavovic, R. Ravishankar, Y. Gandelsman, C. Feichtenhofer, and J. Malik. An Empirical Study of Autoregressive Pre-training from Videos, Jan. 2025.

[90] B. Weng. Navigating the Landscape of Large Language Models: A Comprehensive Review and Analysis of Paradigms and Fine-Tuning Strategies, Apr. 2024.

[91] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer Normalization, July 2016.

[92] B. Zhang and R. Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems 32*, Vancouver, Canada, 2019.

[93] D. Hendrycks and K. Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units, 2017.

[94] R. Sennrich, B. Haddow, and A. Birch. Neural Machine Translation of Rare Words with Subword Units, June 2016.

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

[96] 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)*, pages 2273–2280. IEEE, 2022.

[97] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[98] 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, Sept. 2024.

VERTIFORMER: A Data-Efficient Multi-Task Transformer on Vertically Challenging Terrain

Supplementary Material

The structure of the Appendix is as follows: we start by discussing additional experiments in Section 8, then we will give details of the VERTIFORMER's architecture in Sec. 9, followed by dataset description, implementation details of the robot, FKD, and IKD in Sec. 10, and finally, we provide more qualitative results to showcase VERTIFORMER's performance in Sec. 11.

8 Additional Experiments

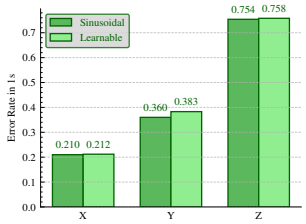
Figure 7: Unseen Test Environments with Rocks/Boulders, Wooden Planks, AstroTurf, and Expanding Foam.

We conduct extensive experiments to demonstrate the efficacy of various features of VERTIFORMER to allow it to be trained with only one hour of data. We also present our findings in a way that highlights VERTIFORMER's differences compared to common practices in NLP and CV, where Transformer training practices have been extensively studied [76, 77, 78, 79, 80, 81, 82]. Therefore, our experiment results also serve as a guideline on how to optimize Transformer training for robotics, particularly in off-road navigation and mobility tasks with complex vehicle-terrain interactions under data-scarce conditions.

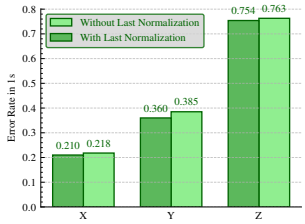
We conduct our experiments based on three perspectives: Section 8.1 provides an analysis of basic factors to train Transformers in general; Finally, Sec. 8.2 evaluates the effectiveness of each off-road mobility learning objective and compares TransformerEncoder, TransformerDecoder, and non-Transformer end-to-end model performances. For fairness, all experiments are conducted with the same hyper-parameters. Please refer to Appendix 8 for the discussions.

Transformers in NLP and CV. The Transformer architecture originated from the seminal work of Vaswani et al. [39] in machine translation. Subsequent research has explored the effects of different Transformer parts, including using only the TransformerEncoder (BERT [18]) or TransformerDecoder (GPT series [17, 19, 20]). Other works explored optimization techniques such as adopting a warm-up phase for training Transformers [76], specific initialization and optimization methods to train deep Transformers with limited data [77], as well as normalization techniques [78].

Early explorations of Transformers in CV include iGPT [83]. A significant breakthrough came with the introduction of Vision Transformers (ViT) by Dosovitskiy et al. [40]. Subsequent research



(a) Positional Encoding



(b) Normalizing Output

Figure 8: (a) Sinusoidal positional encoding achieves better model accuracy than learnable encoding for predicting $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ components of the robot pose. (b) Normalizing the Transformer output before passing the embeddings to the task decoder improves model performance.

focused on refining training methodologies and enhancing performance, such as incorporating auxiliary tasks [80] for spatial understanding, two-stage training (self-supervised view prediction followed by supervised label prediction) [81], different token representations [84], architectural modifications [85], working in embedding space by JEPA family [86, 87, 88], data augmentation and regularization [82], and Masked Autoencoders [21] with random patch encoding for training stabilization [79]. Similar to the autoregressive nature of NLP tasks, Rajasegaran et al. [89] provided empirical guidelines to train Transformers on large-scale video data. Despite the plethora of NLP and CV Transformers trained with internet-scale datasets, existing common training practices may not apply to robot learning with small real-world data, especially for off-road robot mobility.

8.1 Experiment Results of Basic Transformer Factors

Positional encoding is crucial for addressing the permutation equivariance of Transformers, which, by design, lacks inherent sensitivity to input sequence order. This characteristic necessitates the explicit provision of positional information to enable the model to effectively process sequential data. Learnable positional encodings, typically implemented as trainable vectors added to input embeddings, have found favor in CV applications [21]. Conversely, non-learnable encodings, such as the sinusoidal functions introduced in the seminal work by Vaswani et al. [39], have demonstrated efficacy in NLP tasks. This divergence in methodological preference may stem from inherent differences in the statistical properties of data modalities. CV tasks often involve spatially structured data where absolute positional information may be less critical than relative relationships between local features. In such contexts, learnable encodings may offer greater flexibility in adapting to task-specific positional dependencies. Conversely, NLP tasks frequently rely on precise word order and long-range dependencies, where the fixed nature of non-learnable encodings may provide a beneficial inductive bias [90].

To empirically investigate the relative merits of these approaches on robot mobility tasks, we conduct a comparative analysis of learnable positional encodings against sinusoidal encodings as shown in Fig. 8a. Our findings indicate that while both methods achieve comparable asymptotic performance levels, sinusoidal positional encodings exhibit a slight performance advantage.

Normalization layers, such as LayerNorm [91] or RMSNorm [92], have been shown to play a crucial role in stabilizing the training of Large Language Models (LLMs) [78]. By normalizing the activations of hidden units, these layers help to address issues such as vanishing/exploding gradients and improve the overall stability of the training process [76]. In this study, we investigate the impact of applying RMSNorm layer immediately before the task head.

Our experiment results, depicted in Fig. 8b, demonstrate an advantage for a model incorporating RMSNorm layer before the task head. This configuration consistently exhibits improved generalization performance and enhanced training stability compared to a model without the final RMSNorm. This finding suggests that normalizing the final embedding vector before passing it to the task head

can benefit model performance, potentially by facilitating more effective gradient flow and thus improving the robustness of the model's predictions.

8.2 Experiment Results of Robotic Objective Functions

Patch prediction head, as an auxiliary head to learn environment kinodynamics, was first introduced by Nazeri et al. [47]. However, we find that the high complexity of off-road terrain topography and the potential presence of noise or occlusion within the input data create a challenging reconstruction task (see Fig. 1). Consequently, the patch prediction head often generates inaccurate reconstructions, introducing noise into the learning process and negatively impacting the performance of the primary tasks, i.e., FKD, IKD, and BC. This suggests that the auxiliary task of patch reconstruction, in this specific domain, may introduce a conflicting learning signal that hinders the model's ability to effectively learn the desired representations for the main objectives (Fig. 9).

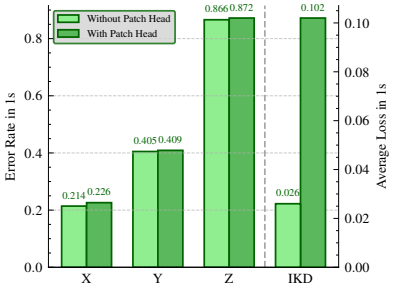


Figure 9: The inclusion of a patch reconstruction head results in a degradation of overall model performance. This counterintuitive result can be attributed to the inherent difficulty in accurately predicting the detailed structure of off-road terrain topography.

9 Model Architecture

Table 2: VERTIFORMER Architecture Parameters.

VERTIENCODER	
Layers	6
Normalization	RMSNorm [92]
Hidden size D	512
Heads	8
MLP size	512
Dropout	0.3
Activation	GELU [93]
Pre-Norm	True
PositionalEncoding	Sinusoidal
VERTIDECODER	
Layers	4
Normalization	RMSNorm [92]
Hidden size D	512
Heads	8
MLP size	512
Dropout	0.3
Activation	GELU [93]
Pre-Norm	True
PositionalEncoding	Sinusoidal

Table 3: End2End Architecture Parameters.

End2End	
Patch Encoder	Resnet-18
Normalization	batch norm [94]
Hidden Layer 1	256
Hidden Layer 2	512
Hidden Layer 3	64
Activation	Tanh
Dropout	0.2

Optimization: we use the AdamW optimizer [95] with learning rate of $5e^{-4}$ and weight decay of 0.08. We train VERTIFORMER for 200 epochs with a batch size of 512. We choose ResNet-18 for End2End model to balance performance with the computational constraints of our robotic platform, making it well-suited for deployment on robots with limited on-board processing capabilities, compared to deeper networks like ResNet-50 or ResNet-101. However, more complex models might offer higher accuracy on the train dataset, it is shown that it is not the case on unseen data [47] since only going deeper does not help understand the intricate interactions between the robot and the terrain.

10 Implementation Details

Dataset: We utilize the dataset introduced by TAL [46], which was collected on a 3.1 m $\times$ 1.3 m modular rock testbed with a maximum height of 0.6 m. The dataset includes 30 minutes of data from both a planar surface and the rock testbed, capturing a diverse range of 6-DoF vehicle states. These states encompass scenarios such as vehicle rollovers and instances of the vehicle getting stuck, all recorded during manual teleoperation over the reconfigurable rock testbed. The dataset comprises visual-inertial odometry for vehicle state estimation, elevation maps derived from depth images, and teleoperation control data, including throttle and steering commands, to provide a holistic view of vehicle dynamics.

10.1 On-Robot Implementation

Hardware: We use an open-source V4W robotic platform, as detailed by Datar et al. [3], for physical evaluation. The V4W platform is equipped with a Microsoft Azure Kinect RGB-D camera to build elevation maps [96] and an NVIDIA Jetson Xavier processor for onboard computation. The proposed VERTIFORMER model is implemented using PyTorch and trained on a single NVIDIA A5000 GPU with 24GB of memory, demonstrating efficient memory utilization with a peak memory footprint of only 2GB.

10.2 Downstream Implementation and Metrics

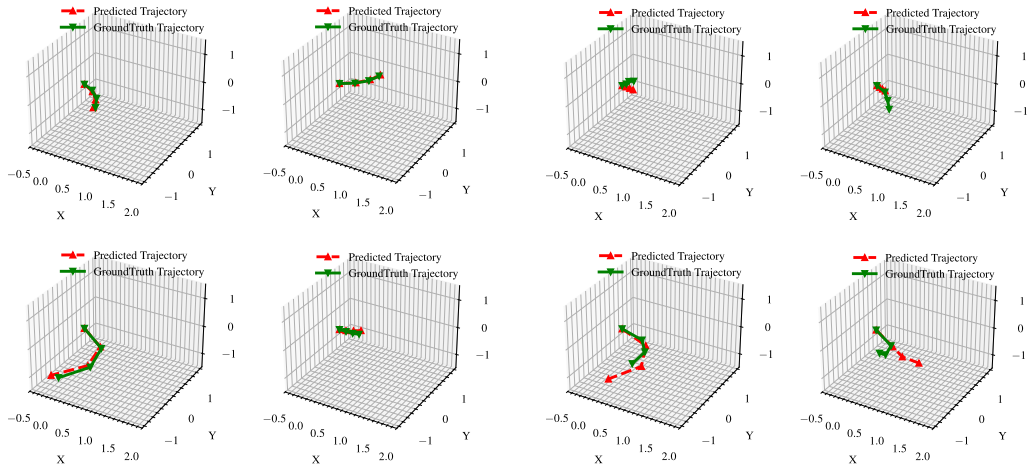
FKD VERTIFORMER's FKD task is integrated with the MPPI planner [74] with 1000 samples and a horizon of 18 steps. We sample across a range of control sequences centered around the last optimal control sequence selected by the robot. The first three actions in a sampled control sequence are passed to VERTIFORMER along with six past poses, actions, and terrain patches at 3 Hz consisting of one second. The model is repeated six times and outputs 18 future poses of the robot, which are combined to create one candidate trajectory. All 1000 candidate trajectories are then evaluated by a cost function, which calculates the cost of each trajectory based on the Euclidean distance to the goal and roll and pitch angles of the robot. Higher distance, roll, and pitch values are penalized with higher cost. Based on the cost function, MPPI outputs the best control sequence moving the robot forward at 3 Hz. The V4W executes the first action and replans.

IKD We integrate VERTIFORMER's IKD task with a global planner based on Dijkstra's algorithm [97], which minimizes traversability cost on a traversability map [98]. The global planner generates three desired future poses with the lowest cost and passes them to VERTIFORMER, which also has access to six past poses, actions, and terrain patches. VERTIFORMER then produces three future actions to drive the robot to the three desired future poses. Similarly to FKD, the V4W executes the first action and then replans at 3 Hz.

NP We implement VERTIFORMER's NP by passing in six past poses, actions, and terrain patches to VERTIFORMER. The model outputs three future actions to take. Similarly to FKD and IKD, the first action is executed by V4W and the replanning of NP runs at 3 Hz.

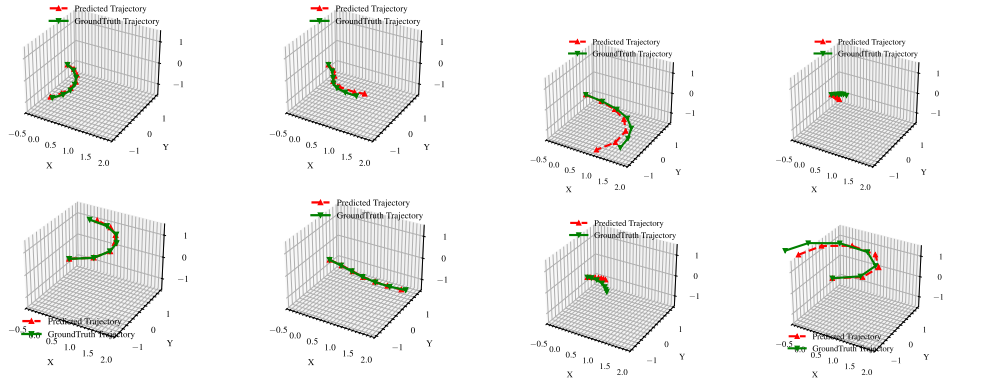
For FKD and IKD, a trial is deemed successful if the robot reaches the defined goal without rolling over or getting stuck. For NP without explicit goal information, a trial is considered successful if the robot successfully traverses the entire testbed.

11 Qualitative Results



(a) Successful 3-step predictions.

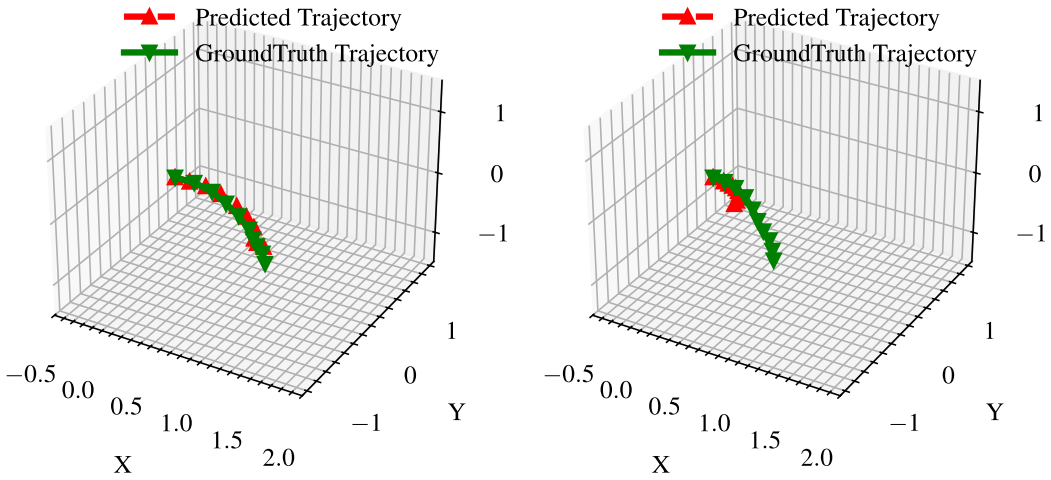
(b) Failed 3-step predictions.



(c) Successful 6-step predictions.

(d) Failed 6-step predictions.

Figure 10: Qualitative Results of 3-Step and 6-Step Successful and Failed Trajectory Prediction over One and Two Second(s).



(a) VERTIFORMER maintains accuracy for longer horizons due to non-autoregressive predictions.

(b) VERTIDECODER drifts from the ground truth due to accumulation of error in autoregressive predictions.

Figure 11: Qualitative Comparison of Drifting between Non-Autoregressive VERTIFORMER and Autoregressive VERTIDECODER.

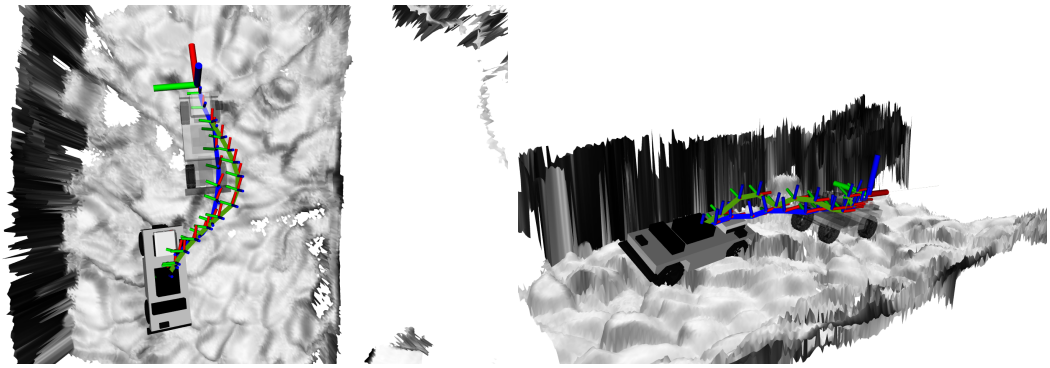


Figure 12: Visualization of VERTIFORMER Predictions in green and Ground Truth in blue.