OFFROAD-VLA: TERRAIN-AWARE ACTION TOKENS AND AFFORDANCE REASONING FOR OFF-ROAD VLA NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models have recently gained significant attention in autonomous driving for their ability to leverage world knowledge for navigation. However, existing VLA approaches primarily target on-road environments, where physical actions assume structured road surfaces and traffic semantics. We present OffRoad-VLA, an end-to-end autoregressive VLA framework that unifies terrain-aware reasoning and physical action generation for off-road navigation. We introduce a terrain-aware physical action codebook that jointly discretizes vehicle motion with local terrain properties, including slope, roughness, and traversability, enabling the model to generate kinematically feasible and terrain-adaptive trajectories. To support efficient reasoning, we employ dual thinking modes that produce direct actions for simple scenes and adaptive thinking using chain-of-thought (CoT) reasoning for complex scenario, explicitly connecting terrain conditions and obstacles to driving actions. We further apply Group Relative Policy Optimization (GRPO) with off-road safety rewards to improve planning while penalizing traversability violations, unsafe terrain interactions, encouraging progress, and smooth motion. Experiments on RELLIS-3D and TartanDrive, together with zero-shot evaluation on the unseen GOOSE dataset, demonstrate improved long-horizon trajectory prediction, terrain classification, and safety over existing approaches. Closed-loop evaluation in the MAVS physics-based simulator further demonstrates robust real-time navigation across diverse off-road conditions.

1 INTRODUCTION

Autonomous navigation in off-road environments presents a fundamental challenge compared to structured urban driving (Seo et al., 2024). Recently, interest for autonomous driving in on-road environments has gain significant attention (Chib & Singh, 2023; Chen et al., 2024a), although a critical gap is still present in case of off-road navigation. Advancement in VLMs has enabled driving systems to incorporate world knowledge and reasoning into its planning loop (Zhou et al.; Tian et al., 2025; Hwang et al.; Shao et al., 2024a). Building on this, VLA models extend these capabilities to direct physical action generation by bridging semantic scenes and low-level trajectory planning (Zhou et al.; Kim et al., 2024; Hirose et al., 2025). These aforementioned VLA models demonstrated superior performance on on-road benchmarks (Caesar et al., 2021; Qian et al., 2024; Dauner et al., 2024). Therefore, their applications are limited to structured on-road navigation only.

Recent advances in off-road autonomous navigation have led to strong terrain adaptive planning methods, including self-supervised traversability learning (Karnan et al., 2023; Sivaprakasam et al., 2024), roughness estimation (Sivaprakasam et al., 2024; Castro et al., 2023), and kinodynamic planning (Damm et al., 2023). Together, these works show that the incorporation of terrain specific properties can significantly improve safety in unstructured environments. However, most of these approaches still depend on classical planners and do not exploit the reasoning capabilities of VLMs. Some recent studies explored VLMs for outdoor navigation (Elnoor et al., 2025; Sathyamoorthy et al., 2024; Shek et al., 2024; Weerakoon et al., 2025), to generate a set of waypoints instead generating full end-to-end trajectories. Although the gap between adaptive planning and VLA-based reasoning remains unresolved, it is essential for off-road navigation on different demographic surfaces.

To address these unresolved problem, we present OffRoad-VLA, a novel end-to-end VLA framework designed specifically for off-road navigation. OffRoad-VLA integrates terrain-aware physical action

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

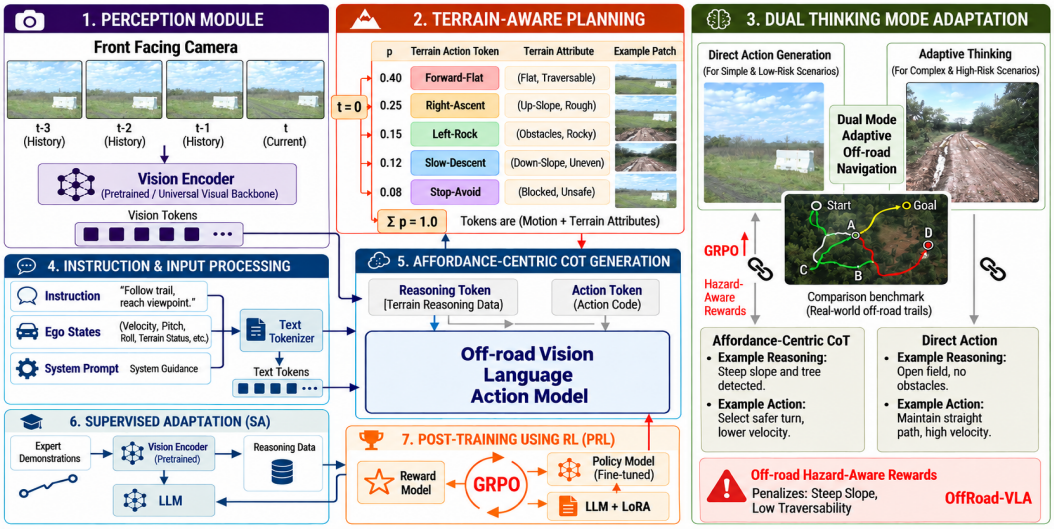


Figure 1: OffRoad-VLA workflow. The framework processes front-view observations through a vision encoder ①, generates terrain-aware action tokens ②, and adaptively selects direct actions or adaptive thinking ③. Language instructions, ego states, and system prompts are integrated as input tokens ④, while affordance-centric CoT connects terrain properties to actions ⑤. The model is trained through supervised adaptation (SA) ⑥ and refined with GRPO-based post-training reinforcement learning (PRL) using hazard-aware rewards ⑦.

tokens directly into a pre-trained VLM backbone, enabling autoregressive planning that jointly optimizes trajectory generation and terrain safety, as shown in Fig. 1. The proposed architecture effectively combines semantic reasoning with terrain-aware physical action generation. During supervised adaption (SA), we use both trajectory and CoT reasoning data to equip the model with dual-process capabilities, supporting both Direct Action Generation (DAG) and Adaptive Thinking (AT). We further perform post-training using reinforcement learning (PRL) with off-road planning reward functions constructed from human feedback (Ouyang et al., 2022) and Group Relative Policy Optimization (GRPO) (Shao et al., 2024b). This proposed design enables terrain adaptive reasoning that balances planning accuracy and efficiency under diverse terrain conditions. Moreover, PRL improves both planning performance and runtime efficiency by reducing unnecessary reasoning.

We evaluate the proposed OffRoad-VLA model on real-world off-road datasets, including RELLIS-3D (Jiang et al., 2021) and TartanDrive (Sivaprakasam et al., 2024), and assess its generalization to unseen conditions using GOOSE (Mortimer et al., 2024) dataset. Moreover, we perform end-to-end closed-loop off-road autonomous driving in the Mississippi State University Autonomous Vehicle Simulator (MAVS) (Hudson et al., 2020; Goodin et al., 2022; Carruth et al., 2024), which is well suited for evaluating robust off-road navigation under different environment. The experimental results show that OffRoad-VLA achieves superior performance across multiple end-to-end off-road autonomous driving benchmarks in both open-loop and closed-loop settings. Our proposed method improves terrain safety and understanding while maintaining competitive planning performance. The main contributions of this paper are summarized as follows:

1. We present OffRoad-VLA, a novel end-to-end VLA framework for unstructured off-road navigation that enables terrain-aware perception, reasoning, and action generation in diverse off-road environments.
2. We propose a terrain-aware physical action codebook that jointly encodes motion kinematics and terrain attributes, enabling safe trajectory generation within a pretrained VLM backbone.
3. OffRoad-VLA introduces affordance-centric CoT reasoning with structured prompts that reasons over terrain and slope types, aligning semantic reasoning with physical constraints.
4. Furthermore, we introduce terrain-aware PRL that reduces reasoning length while improving planning efficiency, enabling the model to generate direct actions when extended reasoning is unnecessary.

2 Related Works

End-to-End VLA-based Autonomous Driving: Now a days, the VLA-based navigation has gain significant advancement that map raw sensor inputs directly to driving actions (Hu et al., 2023; Jiang et al., 2023; Sun et al., 2025). However, the modular driving systems offer interpretability as planners, which suffer from error accumulation and joint optimization (Shao et al., 2023; Jia et al., 2023). Therefore, the end-to-end approaches address these limitations by fine-tuning the VLA models on expert demonstrations. Methods, like UniAD (Hu et al., 2023) and VAD (Jiang et al., 2023), unify perception, prediction, and planning in a single Transformer architecture. The generative methods such as, DiffusionDrive (Liao et al., 2025) and GenAD (Zheng et al., 2024) further improve trajectory diversity and its continuity. Thus, integration of VLMs has opened new directions for understanding semantic scenes and reasoning-guided planning in autonomous driving. Earlier works explored VLMs using captioning (Jin et al., 2023; Park et al., 2024; Ding et al., 2025) and visual question answering (Marcu et al., 2024; Qian et al., 2024), where the reasoning improves adaptability of ego-vehicle in complex scenarios. Subsequent works used VLMs to produce high-level meta actions and driving decisions that supervise downstream planners (Jiang et al., 2024; Wang et al., 2023; Jiang et al., 2025). In recent work, VLA models directly integrate action generation into the VLM backbone to enable end-to-end trajectory planning from raw images and language instructions (Zhou et al.; 2026; Devarakonda et al., 2024; Wang et al., 2024). Another, reasoning-based VLA model such as Alpamayo (Wang et al., 2025b), combines Chain-of-Causation reasoning with diffusion-based trajectory generation and supervised fine-tuning followed by RL. However, mostly existing VLA approaches remains designed for on-road driving. Therefore, the existing methods do not explicitly encode terrain properties in the action space.

Language-Guided Off-Road Navigation: Over the past few years, VLA models have been explored for outdoor navigation, with methods like DriveGPT4 (Xu et al., 2024), MetaVQA (Wang et al., 2025a), OFF-EMMA (Zhang et al., 2026), VAMOS (Castro et al., 2025), GPT-Driver (Mao et al., 2023a), LanguageMPC (Mao et al., 2023b), Driving with LLMs (Chen et al., 2024b), VLM-GroNav (Elnoor et al., 2025), CoNVOI (Sathyamoorthy et al., 2024), LANCAR (Shek et al., 2024), Behav (Weerakoon et al., 2025), OmniVLA (Hirose et al., 2025), and CityWalker (Liu et al., 2025), demonstrating the potential of LLMs and VLMs as interpretable planners and decision-makers. However, these approaches mainly produce high-level commands or waypoints rather than reference trajectories. Thus, they cannot jointly learn action generation with language-guided reasoning. In parallel, safe off-road navigation estimates terrain traversability from robot vision and IMU signals using classical LiDAR-based methods (Frey et al., 2024; Sivaprakasam et al., 2024) and learning-based approaches, including STERLING (Karnan et al., 2023), AirLab (Triest et al., 2023), and *How Does It Feel* (Castro et al., 2023). Moreover, KEASL (Damm et al., 2025) and TRADYN (Achterhold et al., 2024) extend this idea to terrain-aware kinodynamic planning where these methods typically generates cost map and trajectory commands. A recent study (Peng et al., 2026) identified the lack of self-reflective reasoning before execution as an open challenge in VLA-based navigation. Our proposed framework addresses this gap in off-road settings by incorporating terrain information into physical action tokens and enabling reasoning to connect terrain conditions with trajectory planning.

Trajectory Tokenization and Supervised Adaptation for VLA: VLA-based driving requires discretizing continuous trajectories into tokens compatible with VLM decoders. Recent tokenization methods, such as RT-1 (Brohan et al., 2022) uses uniform action bins, while TrajTok (Zheng et al., 2026) use data-driven clustering to capture trajectory distributions. FAST (Pertsch et al., 2025) uses a DCT-based representation for compact trajectory encoding but produces variable-length sequences that complicate fixed-horizon planning. However, these existing tokenizers mostly encode kinematic motion. Although, some work incorporates terrain-aware primitives as continuous cost (Din et al., 2025). On contrary, our action codebook extends K-Disk clustering to jointly encode motion and terrain attribute. RL has also been used to align VLA models with task-specific objectives. DeepSeek-R1 (Guo et al., 2025) demonstrated GRPO-based reasoning optimization, while Gen-Drive (Huang et al., 2025), TrajHF (Li et al., 2025a), RAD (Gao et al., 2025), AutoVLA (Zhou et al.), and AlphaDrive (Jiang et al., 2025) applied RL to safety, preference, and planning objectives. However, these approaches primarily target structured scenarios and do not explicitly address off-road hazards. Therefore, our safety-aware PRL formulation addresses these limitations directly, encouraging fast decisions on safe terrain and deeper reasoning in high-risk scenarios.

Off-Road Datasets and Benchmarks: Evaluation for off-road navigation requires datasets with ground truth terrain annotations and vehicle trajectories. RELLIS-3D (Jiang et al., 2021), an off-road

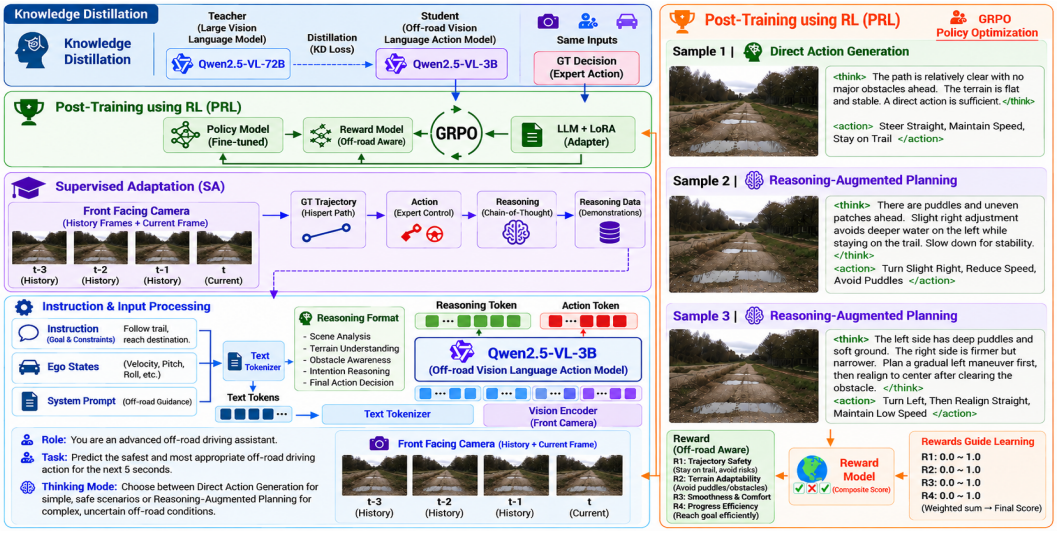


Figure 2: Dual process training of OffRoad-VLA: Training consists of two stages. First, SA distills knowledge from Qwen2.5-VL-72B into the compact Qwen2.5-VL-3B student model using knowledge distillation (KD) loss. The student learns from expert demonstrations pairing camera history with ground-truth trajectories, CoT reasoning, and action tokens. Second, PRL refines the policy using GRPO. The optimization is guided by an off-road aware reward model that evaluates trajectory safety, terrain adaptability, smoothness, and progress efficiency.

dataset, supports terrain traversability (Seo et al., 2023; Bu et al., 2025) and navigability (Guan et al., 2022). Similarly, TartanDrive (Triest et al., 2022) provides multimodal data, including slope and roughness. Another dataset, GOOSE (Mortimer et al., 2024) provides panoptic terrain labels and evaluates navigation using the traversability violation rate. Prior work on outdoor navigation can be grouped as follows: ① Visual navigation foundation models such as GNM (Shah et al., 2023a), ViNT (Shah et al., 2023b), and NoMaD (Sridhar et al., 2024), learn goal-conditioned control directly from images across robot platforms. However, these models consider environment as an obstacle space and therefore they do not explicitly represent terrain traversability. ② Terrain understanding methods, such as Sterling (Karnan et al., 2023), and Terrax (Li et al., 2025b) capture terrain properties, although they leave action generation to a separate planner. Thus, terrain understanding and action selection remain largely decoupled.

3 Methodology

OffRoad-VLA uses a single autoregressive Transformer decoder to jointly generate reasoning and terrain-aware action tokens from front-view images, ego states, and language instructions. As shown in Figure 2, training follows two stages: supervised adaptation (SA), which learns from trajectory data and distills terrain-grounded reasoning from a large VLM, and post-training with reinforcement learning (PRL), which optimizes task-specific rewards for improved planning. The details of OffRoad-VLA model are discussed consequently.

3.1 Problem Formulation

At each timestep t the Offroad-VLA receives camera observations, $\mathbb{C} = \{c_{t-3}, c_{t-2}, c_{t-1}, c_t\}$ capturing $N_f = 4$ sequential frames at 2 Hz to encode scene dynamics. The model additionally receives a high-level navigation instruction $\mathbb{I}$ (such as, go straight or turn left with deceleration) and an ego vehicle state vector $\mathbb{S} = [v, a, \omega, \mathbf{h}]$ where v denotes current speed, a acceleration, ω angular velocity, and $\mathbf{h}$ an action history. Given these inputs, the model produces a sequence of terrain-aware physical action tokens $\mathbf{a} = [a_1, a_2, \ldots, a_T], a_t \in \mathbb{A}$ where $\mathbb{A}$ is action codebook. These tokens decode into a planning trajectory $\mathbb{P} \in \mathbb{R}^{T\times 3}$ over a horizon of T steps. The generated trajectory is both physically feasible and terrain-safe. For complex scenarios, the action sequence is preceded by CoT reasoning sequence $\mathbf{l} = [l_1, \ldots, l_L]$. This reasoning justifies the planned action in terms of

terrain affordances. The full conditional distribution over outputs is defined as,

$$p_\theta(\mathbf{l}, \mathbf{a} \mid \mathbb{C}, \mathbb{I}, \mathbb{S}) = \prod_{i=1}^{L} p_\theta(l_i \mid l_{<i},\, \mathbb{C}, \mathbb{I}, \mathbb{S}) \cdot \prod_{j=1}^{T} p_\theta(a_j \mid \mathbf{l},\, a_{<j},\, \mathbb{C}, \mathbb{I}, \mathbb{S}) \tag{1}$$

where θ denotes the learning parameter of the autoregressive model. We adopt Qwen2.5-VL-3B (Bai et al., 2025) as VLM backbone of OffRoad-VLA. Qwen2.5-VL is a series of powerful multimodal LLMs that possess strong visual understanding capabilities.

3.2 Terrain-Aware Physical Action Tokenization

The key challenge of OffRoad-VLA is the generation of terrain-aware physical action codebooks that jointly discretize vehicle motion kinematics and local terrain properties into a unified set of discrete tokens. Therefore, we discretize continuous trajectories ($\mathbb{P}$) into discrete tokens $\mathbf{a} = [a_1, \ldots, a_T]$ where $T = 10$ motion primitives of $\Delta t = 0.5\,\text{s}$ and each token $a_t \in \mathbb{A}$ jointly encodes kinematic motion $(\Delta x, \Delta y, \Delta\theta)$ and local terrain properties of the traversed segment. This joint representation allows the VLM to simultaneously reason about vehicle dynamics and terrain safety during trajectory planning. We build one terrain-aware physical action codebook per datasets (Jiang et al., 2021; Triest et al., 2022) using an extended K-Disk clustering procedure (Zhang et al., 2025; Wu et al., 2024). Following approaches in (Brohan et al., 2022; Pertsch et al., 2025; Zheng et al., 2026) tokens are incorporated into the VLM vocabulary as special tokens ($\langle\texttt{action}_0\rangle, \langle\texttt{action}_1\rangle, \ldots, \langle\texttt{action}_{K-1}\rangle$). During inference, the model autoregressively generates action token sequences that decode into physically feasible and terrain-safe trajectories using the earlier built codebook. Details of terrain-aware physical action tokenization method are discussed in Appendix A.

3.3 Affordance-Centric Chain-of-Thought Reasoning

Reasoning data provides the high-quality chain-of-thought (CoT) annotations needed to train VLMs with reasoning capabilities (Fu et al., 2025). In on-road autonomous driving, such datasets are well established and have been shown to improve both interpretability and planning quality (Nie et al., 2024; Wu et al., 2025). However, off-road navigation presents a different setting. Therefore, safe actions depend on terrain properties such as surface affordance, slope, roughness, and their interaction with vehicle dynamics (Nie et al., 2024). To the best of our knowledge, no existing reasoning corpus explicitly models this form of terrain-grounded causation. Building such a corpus directly is challenging for three reasons: ① limited coverage of diverse terrain types and driving conditions, ② insufficient representation of physical cues such as terrain affordance, slope, and surface roughness, and ③ low-quality reasoning that does not establish a clear causal connection between terrain conditions and driving actions.

To address these challenges, we develop an automated reasoning annotation pipeline based on the Qwen2.5-VL-72B model (Bai et al., 2025). The pipeline generates structured, terrain-grounded reasoning annotations and enables knowledge distillation from the large teacher model to a compact target model. We provide the ground-truth driving action as an input hint to guide the model toward explanations that explicitly connect the observed terrain to the selected action. Using this pipeline, we annotate approximately 30% of RELLIS-3D (Jiang et al., 2021) and TartanDrive (Triest et al., 2022) training split. We label the remaining segments as direct-decision scenarios. Therefore it allows the model to learn when terrain reasoning is necessary. Further details are provided in Appendix B.

3.4 Supervised Adaptation

In SA, the model is trained to jointly generate terrain-aware reasoning and action tokens from off-road driving data. Given the sensor inputs (as mentioned in section 3.1), the model produces output sequence $\mathbf{x} = [l_1, \ldots, l_L, a_1, \ldots, a_T]$ with L reasoning language tokens and T terrain-aware action tokens. To support both DAG and bothAT during SA, we construct training samples with ground-truth assistant responses in two modes. In DAG mode, the response contains only the final action tokens, preceded by a short template indicating that reasoning is unnecessary. However, in AT mode, the response begins with a template that signals the need for reasoning, followed by structured affordance-centric CoT reasoning and the corresponding action tokens. The language modeling loss $\mathbb{L}_{\text{LM}}$ and auxiliary action loss $\mathbb{L}_{\text{AA}}$ are defined as

where $N = L + T$ and $\mathbb{L}_{\text{LM}}$ applies standard causal language modeling over the full sequence, whereas $\mathbb{L}_{\text{AA}}$ emphasizes planning accuracy on action token positions $\{x_{L+1}, \ldots, x_{L+T}\}$. Moreover,

$$\mathbb{L}_{\mathrm{LM}} = -\frac{1}{N}\sum_{i=1}^{N}\log p_\theta(x_i \mid x_{<i},\, \mathbb{C}, \mathbb{I}, \mathbb{S})$$
$$\mathbb{L}_{\mathrm{AA}} = -\frac{1}{T}\sum_{i=L+1}^{L+T}\log p_\theta(x_i \mid x_{<i},\, \mathbb{C}, \mathbb{I}, \mathbb{S}) \tag{2}$$

$$w_i = \begin{cases} \lambda_{\mathrm{cot}} \cdot \lambda_t & \text{if } \textit{CoT} \;\&\; t_{\mathrm{class}} \in \{2,\, 3\} \\ \lambda_{\mathrm{cot}} & \text{if } \textit{CoT} \;\&\; t_{\mathrm{class}} \in \{0,\, 1\} \\ \lambda_t & \text{if no } \textit{CoT} \;\&\; t_{\mathrm{class}} \in \{2,\, 3\} \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

off-road training data suffer from inherent class imbalance. This occurs because traversable terrain dominates in trajectory distributions, while steep or non-traversable surfaces are rarely observed and thus significantly underrepresented. To prevent the model bias towards flat-terrain behavior, we introduce a terrain stratified per-sample weighting w_i in equation 3, where $\lambda_{\mathrm{cot}} > 1$ upweights CoT samples and $\lambda_t > 1$ upweights hazardous terrain classes. Thus, the combined SA loss is defined as

$$\mathbb{L}_i^{\mathrm{SA}} = w_i \cdot (\mathbb{L}_{\mathrm{LM},i} + \lambda_a \cdot \mathbb{L}_{\mathrm{AA},i}) \tag{4}$$

where λ_a controls action supervision. Additional details of SA are discussed in Appendix C.

3.5 Safety-Aware Post-Training using Reinforcement Learning

To further improve OffRoad-VLA and align its behavior with driving requirements and task-specific objectives, we perform safety-aware post-training using reinforcement learning (RL). Therefore it aligns the model's trajectory generation with explicit off-road safety objectives using policy optimization against verifiable hazard-aware reward signals. We employ GRPO (Shao et al., 2024b; Jiang et al., 2025) as the underlying optimization algorithm. Given a scenario input query q, comprising front images, ego state, and navigation instruction, we sample a group of $\mathcal{G}$ candidate output sequences $\mathbb{O} = \{o_1, \ldots, o_{\mathcal{G}}\}$ from the current policy $\pi_{\theta_{\mathrm{old}}}$. The GRPO objective,

$$\mathbb{J}_{\mathcal{G}}(\theta) = \mathbb{E}_{q,\,\{o_i\}\sim\pi_{\theta_{\mathrm{old}}}}\left[\frac{1}{\mathcal{G}}\sum_{i=1}^{\mathcal{G}}\left(\mathbb{J}_i^R - \xi \cdot \mathcal{D}_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\mathrm{ref}})\right)\right] \tag{5}$$

where clipped surrogate objective for sample o_i with group-normalized advantage (A_i), is defined as,

$$\mathbb{J}_i^R = \min(\rho_i A_i,\, \mathrm{clip}(\rho_i, 1-\varepsilon, 1+\varepsilon)\, A_i)\,, \qquad \rho_i = \frac{\pi_\theta(o_i \mid q)}{\pi_{\theta_{\mathrm{old}}}(o_i \mid q)} \tag{6}$$

$$A_i = \frac{r_i - \mu_r}{\sigma_r}, \qquad \mu_r = \frac{1}{\mathcal{G}}\sum_{j=1}^{\mathcal{G}} r_j, \qquad \sigma_r = \sqrt{\frac{1}{\mathcal{G}}\sum_{j=1}^{\mathcal{G}}(r_j - \mu_r)^2} \tag{7}$$

Here, ε controls the clipping range, and π_{ref} is the frozen reference policy from the SA stage. The KL divergence term is computed using the unbiased approximation as,

$$\mathcal{D}_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\mathrm{ref}}) = \frac{\pi_{\mathrm{ref}}(o_i \mid q)}{\pi_\theta(o_i \mid q)} - \log\frac{\pi_{\mathrm{ref}}(o_i \mid q)}{\pi_\theta(o_i \mid q)} - 1 \tag{8}$$

which stabilizes training by preventing large deviations from the supervised policy while permitting meaningful improvement through off-road reward-guided optimization (Detailed in Appendix D).

4 Experimental Result and Discussions

We train and evaluate OffRoad-VLA separately on each platform, since the action codebook is fitted to the motion regime of its dataset (as discussed in Appendix A.3.2). Both RELLIS-3D (Jiang et al., 2021) and TartanDrive (Triest et al., 2022), datasets are preprocessed into 2 Hz trajectory windows using four history frames and a 10-step (5 s) future trajectory. RELLIS-3D (Jiang et al., 2021) is split by the official sequence protocol, providing 1,713 training, 388 validation, and 380 test samples, and is tokenized with a K=256 codebook. Similarly, TartanDrive (Triest et al., 2022) is partitioned using the official trajectory split, producing 15,416 training, 1,802 validation, and 1,803 test samples, tokenized with K=1024. However, we reserve GOOSE (Mortimer et al., 2024) exclusively for unseen domain evaluation and do not use it during training. The current held-out GOOSE split contains 223 validation samples and 70 test samples. For closed-loop experiments, we use MAVS as the simulation platform. Additional implementation details are provided in Appendix E.

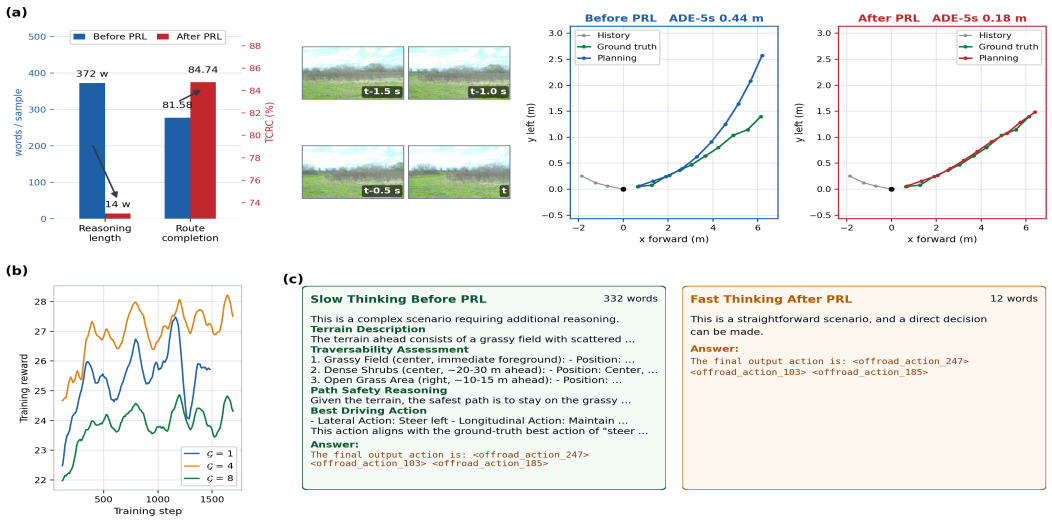


Figure 3: Policy reinforcement learning (PRL) results on the RELLIS-3D dataset. (a) Comparison of reasoning length and TCRC before and after PRL. (b) Training reward curves for different GRPO group sample sizes. (c) qualitative comparison of planning and reasoning behaviour before and after PRL on the same scene.

4.1 In-Distribution Open-Loop Planning and Safety

RELLIS-3D Results: Table 1 presents planning quality comparisons across all methods on the RELLIS-3D dataset. Navigation baselines (Shah et al., 2023a;b; Sridhar et al., 2024; Zhang et al., 2026; Castro et al., 2025) are terrain-agnostic by design and therefore do not report terrain safety metrics. Our SA baseline surpasses all state-of-the-art baselines in terms of trajectory accuracy, while achieving lower terrain-aware safety with a collision rate of 1.32% and TVR of 0.37%. Moreover, OffRoad-VLA with PRL achieves better results across all metrics. The finale trajectory errors over 5 s drop by 0.75 m, with the collision rate reducing to 0.26%, TVR to 0.03%, and TCRC improving to 84.74%. Furthermore, Fig. 3 shows that, terrain-aware PRL improves route completion while substantially reducing reasoning length. The training reward remains stable across different group sizes, with group ($\mathcal{G} = 4$) providing effective policy optimization. The qualitative examples further demonstrate that PRL enables the model to adapt its reasoning depth: it uses detailed reasoning for complex scenes before PRL, while directly generating actions for simpler scenes after PRL. Overall, PRL improves planning efficiency and enables adaptive reasoning based on scene complexity.

Table 1: Comparison of planning quality on RELLIS-3D. **Bold** = best, underline = second best.

Method	Planning Quality						
	ADE-1s ↓	ADE-3s ↓	ADE-5s ↓	FDE-5s ↓	Collision (%) ↓	TVR (%) ↓	TCRC (%) ↑
GNM (Shah et al., 2023a)	0.98	2.07	3.15	5.59	–	–	6.27
ViNT (Shah et al., 2023b)	0.93	2.11	3.21	5.68	–	–	5.76
NoMaD (Sridhar et al., 2024)	1.03	2.44	3.84	6.83	–	–	2.26
OFF-EMMA (Zhang et al., 2026)	0.37	2.16	3.95	5.61	–	–	4.84
VAMOS (Castro et al., 2025)	0.91	1.97	2.43	4.84	–	–	6.77
Alpamayo (Wang et al., 2025b)	0.11	0.45	1.25	3.51	49.74	25.5	19.47
OffRoad-VLA (SA)	0.12	0.26	**0.42**	0.81	1.32	0.37	81.58
OffRoad-VLA (PRL)	**0.11**	**0.25**	**0.42**	**0.75**	**0.26**	**0.03**	**84.74**

TartanDrive Results: Table 2 presents planning quality comparisons on the TartanDrive dataset (Triest et al., 2022). Similar, to RELLIS-3D, the same trends is observed here as well. Trajectory errors in PRL, are substantially reduced compared to SA, with short and mid-horizon errors nearly halved and long-horizon error decreased by 12.24%, while the collision rate drops to 0.67%, TVR to $0.0.07\%$ and TCRC improves to 5.27%. These consistent gains confirm that the PRL is particularly effective under the demanding dynamics of high-speed off-road traversal. A qualitative analysis of action generation across different scenarios in the RELLIS-3D and TartanDrive is presented in Appendix F.1.

Table 2: Comparison of planning quality on TartanDrive. **Bold** = best, underline = second best.

Method	Planning Quality						
	ADE-1s ↓	ADE-3s ↓	ADE-5s ↓	FDE-5s ↓	Collision (%) ↓	TVR (%) ↓	TCRC (%) ↑
GNM (Shah et al., 2023a)	0.96	2.01	3.41	5.93	–	–	0.67
ViNT (Shah et al., 2023b)	1.74	2.17	4.54	7.62	–	–	0.51
NoMaD (Sridhar et al., 2024)	0.63	2.11	3.57	7.04	–	–	0.58
OFF-EMMA (Zhang et al., 2026)	1.58	2.89	4.40	7.2	–	–	0.58
VAMOS (Castro et al., 2025)	0.85	2.21	3.47	7.79	–	–	0.67
Alpamayo (Wang et al., 2025b)	0.43	1.55	3.28	7.61	–	–	0.03
OffRoad-VLA (SA)	<u>0.35</u>	<u>1.23</u>	<u>2.45</u>	**5.43**	<u>1.28</u>	<u>0.13</u>	<u>4.83</u>
OffRoad-VLA (PRL)	**0.32**	**1.11**	**2.15**	<u>5.69</u>	**0.67**	**0.07**	**5.27**

We further evaluate terrain-aware safety planning on RELLIS-3D and TartanDrive, benchmarking against perception-centric planners (detailed results in Appendix F.2).

4.2 Zero-Shot Generalization to Unseen Off-Road Environments

Evaluating generalization beyond the training distribution is critical for real-world deployment. Thus, we use the GOOSE dataset, which was collected in different demographic regions. Table 3 compares OffRoad-VLA's zero-shot performance on GOOSE after training exclusively on RELLIS-3D and TartanDrive with existing state-of-the-art methods. Classical navigation models (Shah et al., 2023a;b; Sridhar et al., 2024; Zhang et al., 2026) fail to generalize the unstructured off-road terrain. These methods produce ADE-5s above 17 m and miss rates exceeding 90 % reflecting their on-road-oriented design. In contrast, OffRoad-VLA (Tartandrive) achieves ADE-5s of 4.46 m and FDE-5s of 8.67 m. This represents a more than three-fold improvement in long-horizon trajectory accuracy. Miss rates improve significantly to MR_2 of 83% and MR_4 of 55.6% confirming substantially closer alignment to the ground-truth trajectories across all prediction horizons.

Table 3: Zero-shot open-loop testing on GOOSE. **Bold** = best, underline = second best.

Method	Planning Quality				Failure Rate		Terrain Safety		
	ADE-1s ↓	ADE-3s ↓	ADE-5s ↓	FDE-5s ↓	MR_2 (%) ↓	MR_4 (%) ↓	Col (%) ↓	TVR (%) ↓	SS ↑
GNM (Shah et al., 2023a)	5.16	11.64	17.85	31.68	90	91.43	–	–	**1.00**
ViNT (Shah et al., 2023b)	5.01	11.5	17.72	31.56	90	90.71	–	–	**1.00**
NoMaD (Sridhar et al., 2024)	5.15	11.71	18.01	31.98	91.43	92.86	–	–	**1.00**
OFF-EMMA (Zhang et al., 2026)	3.47	8.06	12.77	23.52	92.86	93.14	–	–	**1.00**
Terrax (Li et al., 2025b)	–	–	–	–	–	–	0.17	0.28	<u>0.99</u>
Sterling (Karnan et al., 2023)	–	–	–	–	–	–	1.43	2.57	0.82
GPT-DRIVER (Mao et al., 2023a)	–	–	–	–	–	–	0.79	0.31	0.96
MetaVQA (Wang et al., 2025a)	–	–	–	–	–	–	0.25	0.37	0.93
VAMOS (Castro et al., 2025)	1.97	3.07	6.83	16.83	88.57	89.14	<u>0.14</u>	<u>0.23</u>	**1.00**
Alpamayo (Wang et al., 2025b)	**0.67**	**1.79**	<u>5.33</u>	<u>9.19</u>	<u>87</u>	<u>62.78</u>	0.93	0.34	0.69
OffRoad-VLA (RELLIS-3D)	2.12	4.94	7.75	14.04	98.2	89.2	**0.00**	**0.00**	0.92
OffRoad-VLA (TartanDrive)	<u>1.08</u>	<u>2.66</u>	**4.46**	**8.67**	**83.0**	**55.6**	**0.00**	**0.00**	0.85

For terrain safety, OffRoad-VLA (TartanDrive) outperforms all terrain understanding planners (Li et al., 2025b; Karnan et al., 2023), and VLM-based driving models (Wang et al., 2025a; Mao et al., 2023a; Castro et al., 2025; Wang et al., 2025b). Both pretrained OffRoad-VLA (RELLIS-3D and TaratanDrive) achieves the zero collision and terrain violation rate. These results demonstrate that OffRoad-VLA generalizes itself from seen to unseen environments, while maintaining both trajectory accuracy and terrain safety. Qualitative analysis of the zero-shot testing is discussed in Appendix F.3.

4.3 Closed-Loop Evaluation

Table 4 compares OffRoad-VLA against end-to-end navigation models (Shao et al., 2024a; Sima et al., 2024; Xu et al., 2024; Zhou et al.; Zhang et al., 2026). The model is trained using SA and operates at a planning frequency of 2 Hz during testing. OffRoad-VLA achieving highest average speed of 4.31 *m/s*, reflecting a balanced trade-off between traversal efficiency and terrain compliance. Moreover, these closed-loop results confirm that OffRoad-VLA translates its terrain-awareness into end-to-end navigation, achieving the best route completion and success rates in physics-based simulated environments for off-road navigation. Detailed experimental setup is explained in Appendix G.

4.4 Ablation Studies and Analysis

Codebook & Tokenization Quality: Selecting the codebook size requires balancing representation quality and prediction efficiency. As shown in Appendix A.3.2, larger vocabularies reduce reconstruction error although, can reduce planning. We therefore select the vocabulary size based on

Table 4: Closed-Loop evaluation on MAVS.

Models	DS ↑	SR (%) ↑	Avg Speed →
LMDrive (Shao et al., 2024a)	0.29	77.00	1.11
DriveLM (Sima et al., 2024)	0.82	81.23	1.03
DriveGPT4 (Xu et al., 2024)	0.89	<u>83.75</u>	1.19
AutoVLA (Zhou et al.)	0.89	82.14	<u>1.33</u>
OFF-EMMA (Zhang et al., 2026)	**0.93**	81.71	0.82
OffRoad-VLA	<u>0.91</u>	**84**	**4.31**

the available data, using $K{=}256$ for RELLIS-3D and $K{=}1024$ for TartanDrive. At these sizes, we compare our terrain-aware physical action codebook with existing action tokenizers (Pertsch et al., 2025; Brohan et al., 2022; Zheng et al., 2026), as shown in Table 5a.

RT-1 (Brohan et al., 2022) discretizes acceleration and steering rate into uniform bins and reconstructs trajectories using a kinematic model, which introduces errors when recovering controls from trajectory data. Its distribution-agnostic representation achieves only 56.2% coverage on TartanDrive. Conversely, FAST (Pertsch et al., 2025) uses a frequency-domain representation designed for smooth motion. Although, it is less suitable for abrupt off-road heading changes. As a result, produces the highest reconstruction errors and below 6% coverage. However, TrajTok (Zheng et al., 2026) adapts its representation to the data distribution and provides the strongest baseline, achieving full coverage on both datasets. On contrary, 0ur terrain-aware K-Disk achieves the lowest FDE on both datasets and the lowest ADE on RELLIS-3D at 0.0265 m, while remaining within 0.4% of TrajTok on TartanDrive. In addition to accurate reconstruction, our codebook associates each action token with terrain attributes, explicitly linking the maneuver to its terrain context.

Table 5: **(a)** Action tokenizer comparison under per-step rollout reconstruction. **(b)** Influence of terrain-aware physical action tokenization on RELLIS-3D at $K{=}256$.

(a) Action tokenizer comparison

	RELLIS-3D ($K{=}256$)			TartanDrive ($K{=}1024$)		
Tokenizer	ADE↓	FDE↓	CC↑	ADE↓	FDE↓	CC↑
FAST (Pertsch et al., 2025)	0.4858	0.5493	5.9	2.8180	3.4048	4.7
RT-1 bins (Brohan et al., 2022)	0.0528	0.0574	79.3	0.1088	0.1191	56.2
TrajTok (Zheng et al., 2026)	0.0289	0.0308	**100.0**	**0.0427**	0.0462	**100.0**
Ours	**0.0265**	**0.0271**	96.5	0.0431	**0.0445**	**100.0**

(b) Terrain-aware tokenization

Tokenizer	Vocabulary		Planning		Terrain	
	Recon. ADE↓	CC↑	ADE-5s↓	FDE-5s↓	TCA↑	SS↑
Kinematic ($\lambda_\phi{=}0$)	**0.0233**	86.5	0.4809	0.9421	70.32	0.73
Terrain-aware ($\lambda_\phi{=}0.3$)	0.0265	**96.5**	**0.4180**	**0.8145**	**79.76**	**0.85**

Effectiveness of Terrain-aware Tokenization: To isolate the effect of terrain awareness, we compare terrain-aware and kinematics-only codebooks while keeping the clustering procedure, vocabulary size, training data, and hyperparameters same. As shown in Table 5b, incorporating terrain information consistently outperforms kinematics-only tokenization, improving codebook coverage, planning accuracy, terrain classification, and safety, with only a marginal increase in reconstruction error. This demonstrates that terrain-aware tokenization produces more expressive action representations that better support terrain-aware and safer off-road planning.

5 Conclusion

We presented OffRoad-VLA, an end-to-end VLA framework for off-road navigation that encodes vehicle kinematics and terrain attributes into a unified terrain-aware action codebook, vehicle trajectory generation in affordance-centric CoT reasoning. A two-stage training procedure combining SA with safety-aware PRL enables adaptive planning while penalizing traversability violations and slope-induced risk. Evaluation across RELLIS-3D, TartanDrive, and the unseen GOOSE benchmark shows that OffRoad-VLA consistently outperforms navigation and perception-centric LLM baselines on both trajectory accuracy and terrain safety. The proposed method achieves benchmarking token prediction accuracy alongside strictly positive roughness correlation across all datasets. Zero-shot generalization to GOOSE confirms robust transfer to unseen environments and platforms, while closed-loop MAVS simulation validates reliable end-to-end navigation with the best success rate and TCRC rate among all evaluated methods. OffRoad-VLA operates at a 2 Hz planning frequency in closed-loop evaluation, however, its current inference latency remains GPU-intensive and does not yet support 2 Hz real-time onboard execution. Although, PRL substantially reduces the gap and can be further improved through techniques such as quantization and speculative decoding.

AI USE STATEMENT

We used generative AI tools solely to assist with language editing, grammar correction, and improving the clarity and readability of the manuscript. All AI-assisted text was reviewed and edited by the authors for accuracy, consistency, and alignment with the reported experiments. The authors take full responsibility for the final content of this work, including all claims, analyses, results, and artifacts.

ETHICS STATEMENT

This work focuses on autonomous off-road navigation using visual, language, and vehicle-state information. The study uses publicly available datasets, including RELLIS-3D, TartanDrive, and GOOSE, which contain outdoor off-road scenes collected for research purposes. We use these datasets solely for research and evaluation of terrain-aware navigation and action generation. No personally identifiable information is collected, generated, or used by our method. The proposed framework is designed to improve the safety and reliability of autonomous navigation by explicitly modeling terrain properties, traversability, and vehicle motion. All experiments are conducted in controlled simulation or research environments, and the reported results are intended to support responsible development and evaluation of autonomous off-road systems.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the OffRoad-VLA architecture, terrain-aware physical action codebook, adaptive reasoning mechanism, supervised adaptation (SA), and post-training reinforcement learning (PRL) in the main paper and Appendix. We specify the datasets, preprocessing procedures, training configurations, codebook construction, reward functions, evaluation protocols, and metrics used in our experiments. Additional implementation details, hyperparameters, ablation settings, and qualitative results are provided in the Appendix to facilitate reproduction of the reported results. All experiments are conducted using consistent evaluation protocols across RELLIS-3D and TartanDrive, with GOOSE used for zero-shot out-of-distribution evaluation. Closed-loop evaluation is performed using the MAVS physics-based simulator under the same predefined experimental settings.

REFERENCES

Jan Achterhold, Suresh Guttikonda, Jens U Kreber, Haolong Li, and Joerg Stueckler. Learning a terrain-and robot-aware dynamics model for autonomous mobile robot navigation. *arXiv preprint arXiv:2409.11452*, 2024.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Yafeng Bu, Zhenping Sun, Xiaohui Li, Jun Zeng, Xin Zhang, and Hui Shen. Self-supervised traversability learning with online prototype adaptation for off-road autonomous driving. *IEEE Robotics and Automation Letters*, 2025.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

Daniel W Carruth, Christopher Goodin, Lalitha Dabbiru, Nicklaus Scherrer, Marc N Moore, Christopher H Hudson, Lucas D Cagle, and Paramsothy Jayakumar. Comparing real and simulated performance for an off-road autonomous ground vehicle in obstacle avoidance. *Journal of Field Robotics*, 41(3):798–810, 2024.

Mateo Guaman Castro, Samuel Triest, Wenshan Wang, Jason M Gregory, Felix Sanchez, John G Rogers, and Sebastian Scherer. How does it feel? self-supervised costmap learning for off-road vehicle traversability. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 931–938. IEEE, 2023.

Mateo Guaman Castro, Sidharth Rajagopal, Daniel Gorbatov, Matt Schmittle, Rohan Baijal, Octi Zhang, Rosario Scalise, Sidharth Talia, Emma Romig, Celso de Melo, et al. Vamos: A hierarchical vision-language-action model for capability-modulated and steerable navigation. *arXiv preprint arXiv:2510.20818*, 2025.

Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10164–10183, 2024a.

Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14093–14100. IEEE, 2024b.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

Pranav Singh Chib and Pravendra Singh. Recent advancements in end-to-end autonomous driving using deep learning: A survey. *IEEE Transactions on Intelligent Vehicles*, 9(1):103–118, 2023.

Eric R Damm, Jason M Gregory, Eli S Lancaster, Felix A Sanchez, Daniel M Sahu, and Thomas M Howard. Terrain-aware kinodynamic planning with efficiently adaptive state lattices for mobile robot navigation in off-road environments. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9918–9925. IEEE, 2023.

Eric R Damm, Eli S Lancaster, Felix A Sanchez, Kiana Bronder, Jason M Gregory, and Thomas M Howard. Incorporating stochastic models of controller behavior into kinodynamic efficiently adaptive state lattices for mobile robot motion planning in off-road environments. *arXiv preprint arXiv:2508.04384*, 2025.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.

Venkata Naren Devarakonda, Raktim Gautam Goswami, Ali Umut Kaypak, Naman Patel, Rooholla Khorrambakht, Prashanth Krishnamurthy, and Farshad Khorrami. Orionnav: Online planning for robot autonomy with context-aware llm and open-vocabulary semantic scene graphs. *arXiv preprint arXiv:2410.06239*, 2024.

Muhayy Ud Din, Waseem Akram, Lyes Saad Saoud, Jan Rosell, and Irfan Hussain. Multimodal fusion with vision-language-action models for robotic manipulation: A systematic review. *Information Fusion*, pp. 104062, 2025.

Kairui Ding, Boyuan Chen, Yuchen Su, Huan-ang Gao, Bu Jin, Chonghao Sima, Xiaohui Li, Wuqiang Zhang, Paul Barsch, Hongyang Li, et al. Hint-ad: Holistically aligned interpretability in end-to-end autonomous driving. In *Conference on Robot Learning*, pp. 3742–3765. PMLR, 2025.

Mohamed Elnoor, Kasun Weerakoon, Gershom Seneviratne, Ruiqi Xian, Tianrui Guan, Mohamed Khalid M Jaffar, Vignesh Rajagopal, and Dinesh Manocha. Vlm-gronav: Robot navigation using physically grounded vision-language models in outdoor environments. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2391–2398. IEEE, 2025.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9710–9719, 2021.

Jonas Frey, Manthan Patel, Deegan Atha, Julian Nubert, David Fan, Ali Agha, Curtis Padgett, Patrick Spieler, Marco Hutter, and Shehryar Khattak. Roadrunner—learning traversability estimation for autonomous off-road driving. *IEEE Transactions on Field Robotics*, 1:192–212, 2024.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24823–24834. IEEE, 2025.

Hao Gao, Shaoyu Chen, Bo Jiang, Bencheng Liao, Yiang Shi, Xiaoyang Guo, Yuechuan Pu, Haoran Yin, Xiangyu Li, Xinbang Zhang, et al. Rad: Training an end-to-end driving policy via large-scale 3dgs-based reinforcement learning. *arXiv preprint arXiv:2502.13144*, 2025.

Christopher Goodin, Daniel W Carruth, Lalitha Dabbiru, Christopher H Hudson, Lucas D Cagle, Nicklaus Scherrer, Marc N Moore, and Paramsothy Jayakumar. Simulation-based testing of autonomous ground vehicles. In *Autonomous Systems: Sensors, Processing and Security for Ground, Air, Sea and Space Vehicles and Infrastructure 2022*, volume 12115, pp. 167–174. SPIE, 2022.

Tianrui Guan, Divya Kothandaraman, Rohan Chandra, Adarsh Jagan Sathyamoorthy, Kasun Weerakoon, and Dinesh Manocha. Ga-nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments. *IEEE Robotics and Automation Letters*, 7(3):8138–8145, 2022.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Raphael Hagmanns, Peter Mortimer, Miguel Granero, Thorsten Luettel, and Janko Petereit. Excavating in the wild: The goose-ex dataset for semantic segmentation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10378–10386. IEEE, 2025.

Noriaki Hirose, Catherine Glossop, Dhruv Shah, and Sergey Levine. Omnivla: An omni-modal vision-language-action model for robot navigation. *arXiv e-prints*, pp. arXiv–2509, 2025.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17853–17862, 2023.

Zhiyu Huang, Xinshuo Weng, Maximilian Igl, Yuxiao Chen, Yulong Cao, Boris Ivanovic, Marco Pavone, and Chen Lv. Gen-drive: Enhancing diffusion generative driving policies with reward modeling and reinforcement learning fine-tuning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3445–3451. IEEE, 2025.

Christopher R Hudson, Christopher Goodin, Zach Miller, Warren Wheeler, and Daniel W Carruth. Mississippi state university autonomous vehicle simulation library. In *2020 Ground Vehicle Systems Engineering and Technology Symposium*. SAE Technical Paper, 2020.

Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, et al. Emma: End-to-end multimodal model for autonomous driving. *Transactions on Machine Learning Research*.

Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21983–21994, 2023.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8306–8316. IEEE, 2023.

Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. *arXiv preprint arXiv:2410.22313*, 2024.

Bo Jiang, Shaoyu Chen, Qian Zhang, Wenyu Liu, and Xinggang Wang. Alphadrive: Unleashing the power of vlms in autonomous driving via reinforcement learning and reasoning. *arXiv preprint arXiv:2503.07608*, 2025.

Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripalli. Rellis-3d dataset: Data, benchmarks and analysis. In *2021 IEEE international conference on robotics and automation (ICRA)*, pp. 1110–1116. IEEE, 2021.

Bu Jin, Xinyu Liu, Yupeng Zheng, Pengfei Li, Hao Zhao, Tong Zhang, Yuhang Zheng, Guyue Zhou, and Jingjing Liu. Adapt: Action-aware driving caption transformer. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7554–7561. IEEE, 2023.

Haresh Karnan, Elvin Yang, Daniel Farkash, Garrett Warnell, Joydeep Biswas, and Peter Stone. Sterling: Self-supervised terrain representation learning from unconstrained robot experience. In *Conference on Robot Learning*, pp. 2393–2413. PMLR, 2023.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Hojin Lee, Taekyung Kim, Jungwi Mun, and Wonsuk Lee. Learning terrain-aware kinodynamic model for autonomous off-road rally driving with model predictive path integral control. *IEEE Robotics and Automation Letters*, 8(11):7663–7670, 2023.

Derun Li, Changye Li, Yue Wang, Jianwei Ren, Xin Wen, Pengxiang Li, Leimeng Xu, Kun Zhan, Peng Jia, Xianpeng Lang, et al. Learning personalized driving styles via reinforcement learning from human feedback. *arXiv preprint arXiv:2503.10434*, 2025a.

Hongze Li, Xuchuan Huang, Xinhai Chang, Jun Zhou, and Huijing Zhao. Terrax: Visual terrain classification enhanced by vision-language models. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 15383–15390. IEEE, 2025b.

Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14864–14873, 2024.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12037–12047, 2025.

Xinhao Liu, Jintong Li, Yicheng Jiang, Niranjan Sujay, Zhicheng Yang, Juexiao Zhang, John Abanes, Jing Zhang, and Chen Feng. Citywalker: Learning embodied urban navigation from web-scale videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6875–6885, 2025.

Jiageng Mao, Yuxi Qian, Junjie Ye, Hang Zhao, and Yue Wang. Gpt-driver: Learning to drive with gpt. *arXiv preprint arXiv:2310.01415*, 2023a.

Jiageng Mao, Junjie Ye, Yuxi Qian, Marco Pavone, and Yue Wang. A language agent for autonomous driving. *arXiv preprint arXiv:2311.10813*, 2023b.

Ana-Maria Marcu, Long Chen, Jan Hünermann, Alice Karnsund, Benoit Hanotte, Prajwal Chidananda, Saurabh Nair, Vijay Badrinarayanan, Alex Kendall, Jamie Shotton, et al. Lingoqa: Visual question answering for autonomous driving. In *European Conference on Computer Vision*, pp. 252–269. Springer, 2024.

Peter Mortimer, Raphael Hagmanns, Miguel Granero, Thorsten Luettel, Janko Petereit, and Hans-Joachim Wuensche. The goose dataset for perception in unstructured environments. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14838–14844. IEEE, 2024.

Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving. In *European Conference on Computer Vision*, pp. 292–308. Springer, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

SungYeon Park, MinJae Lee, JiHyuk Kang, Hahyeon Choi, Yoonah Park, Juhwan Cho, Adam Lee, and DongKyu Kim. Vlaad: Vision and language assistant for autonomous driving. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 980–987, 2024.

Zhenghao Peng, Wenhao Ding, Yurong You, Yuxiao Chen, Wenjie Luo, Thomas Tian, Yulong Cao, Apoorva Sharma, Danfei Xu, Boris Ivanovic, et al. Counterfactual vla: Self-reflective vision-language-action model with adaptive reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4022–4031, 2026.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Tran Phong, Haoran Wu, Cunjun Yu, Panpan Cai, Sifa Zheng, and David Hsu. What truly matters in trajectory prediction for autonomous driving? *Advances in Neural Information Processing Systems*, 36:71327–71339, 2023.

Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4542–4550, 2024.

Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. {Zero-offload}: Democratizing {billion-scale} model training. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pp. 551–564, 2021.

Adarsh Jagan Sathyamoorthy, Kasun Weerakoon, Mohamed Elnoor, Anuj Zore, Brian Ichter, Fei Xia, Jie Tan, Wenhao Yu, and Dinesh Manocha. Convoi: Context-aware navigation using vision language models in outdoor and indoor environments. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 13837–13844. IEEE, 2024.

Junwon Seo, Sungdae Sim, and Inwook Shim. Learning off-road terrain traversability with self-supervisions only. *IEEE Robotics and Automation Letters*, 8(8):4617–4624, 2023.

Junwon Seo, Taekyung Kim, Seongyong Ahn, and Kiho Kwak. Metaverse: Meta-learning traversability cost map for off-road navigation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 13190–13197. IEEE, 2024.

Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7226–7233. IEEE, 2023a.

Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation. *arXiv preprint arXiv:2306.14846*, 2023b.

Hao Shao, Letian Wang, Ruobing Chen, Steven L Waslander, Hongsheng Li, and Yu Liu. Reasonnet: End-to-end driving with temporal and global reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13723–13733, 2023.

Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15120–15130, 2024a.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv e-prints*, pp. arXiv–2402, 2024b.

Chak Lam Shek, Xiyang Wu, Wesley A Suttle, Carl Busart, Erin Zaroukian, Dinesh Manocha, Pratap Tokekar, and Amrit Singh Bedi. Lancar: Leveraging language for context-aware robot locomotion in unstructured environments. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9612–9619. IEEE, 2024.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pp. 256–274. Springer, 2024.

Matthew Sivaprakasam, Parv Maheshwari, Mateo Guaman Castro, Samuel Triest, Micah Nye, Steve Willits, Andrew Saba, Wenshan Wang, and Sebastian Scherer. Tartandrive 2.0: More modalities and better infrastructure to further self-supervised learning research in off-road driving tasks. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12606–12606. IEEE, 2024.

Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 63–70. IEEE, 2024.

Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8795–8801. IEEE, 2025.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, XianPeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. In *Conference on Robot Learning*, pp. 4698–4726. PMLR, 2025.

Samuel Triest, Matthew Sivaprakasam, Sean J Wang, Wenshan Wang, Aaron M Johnson, and Sebastian Scherer. Tartandrive: A large-scale dataset for learning off-road dynamics models. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2546–2552. IEEE, 2022.

Samuel Triest, Mateo Guaman Castro, Parv Maheshwari, Matthew Sivaprakasam, Wenshan Wang, and Sebastian Scherer. Learning risk-aware costmaps via inverse reinforcement learning for off-road navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 924–930. IEEE, 2023.

Tianqi Wang, Enze Xie, Ruihang Chu, Zhenguo Li, and Ping Luo. Drivecot: Integrating chain-of-thought reasoning with end-to-end driving. *arXiv preprint arXiv:2403.16996*, 2024.

Weizhen Wang, Chenda Duan, Zhenghao Peng, Yuxin Liu, and Bolei Zhou. Embodied scene understanding for vision language models via metavqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22453–22464, 2025a.

Wenhai Wang, Jiangwei Xie, ChuanYang Hu, Haoming Zou, Jianan Fan, Wenwen Tong, Yang Wen, Silei Wu, Hanming Deng, Zhiqi Li, et al. Drivemlm: Aligning multi-modal large language models with behavioral planning states for autonomous driving. *arXiv preprint arXiv:2312.09245*, 2023.

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025b.

Kasun Weerakoon, Mohamed Elnoor, Gershom Seneviratne, Vignesh Rajagopal, Senthil Hariharan Arul, Jing Liang, Mohamed Khalid M Jaffar, and Dinesh Manocha. Behav: Behavioral rule guided autonomy using vlms for robot navigation in outdoor scenes. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7044–7051. IEEE, 2025.

Dongming Wu, Wencheng Han, Yingfei Liu, Tiancai Wang, Cheng-zhong Xu, Xiangyu Zhang, and Jianbing Shen. Language prompt for autonomous driving. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 8359–8367, 2025.

Wei Wu, Xiaoxin Feng, Ziyan Gao, and Yuheng Kan. Smart: Scalable multi-agent real-time motion generation via next-token prediction. *Advances in Neural Information Processing Systems*, 37: 114048–114071, 2024.

Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 9(10):8186–8193, 2024.

Liangdong Zhang, Yiming Nie, Haoyang Li, Fanjie Kong, Baobao Zhang, Shunxin Huang, Kai Fu, Chen Min, and Liang Xiao. A vision-language-action model with visual prompt for off-road autonomous driving. *arXiv preprint arXiv:2601.03519*, 2026.

Zhejun Zhang, Peter Karkus, Maximilian Igl, Wenhao Ding, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Closed-loop supervised fine-tuning of tokenized traffic models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5422–5432, 2025.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

Zhipeng Zhao, Taimeng Fu, Shaoshu Su, Qiwei Du, Ehsan Tarkesh Esfahani, Karthik Dantu, Souma Chowdhury, and Chen Wang. Learning when to jump for off-road navigation. *arXiv preprint arXiv:2602.00877*, 2026.

Chenhao Zheng, Jieyu Zhang, Jianing Zhang, Weikai Huang, Ashutosh Kumar, Quan Kong, Oncel Tuzel, Chun-Liang Li, and Ranjay Krishna. Trajtok: Learning trajectory tokens enables better video understanding. *arXiv preprint arXiv:2602.22779*, 2026.

Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *European Conference on Computer Vision*, pp. 87–104. Springer, 2024.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, Volker Tresp, and Alois Knoll. Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 13782–13790, 2026.

Zewei Zhou, Tianhui Cai, Seth Z Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

A Terrain-Aware Physical Action Tokenization

A.1 Terrain Attribute Encoding

For each trajectory segment of duration $\Delta t = 0.5\,\mathrm{s}$, we extract a terrain attribute vector $\boldsymbol{\phi}_j \in \mathbb{R}^{d_\phi}$ comprising four complementary dimensions that together characterize the geometric, dynamic, and semantic properties of the traversed surface. All four dimensions are derived from proprioceptive signals already present in the driving logs (such as the ego pose and IMU data). These attributes are used during codebook construction and training. Terrain awareness is encoded directly within the action tokens themselves through the clustering process, as detailed subsequently.

Traversability Class: The first dimension is the *traversability class* $t^j_{\text{class}} \in \{0, 1, 2, 3\}$ a discrete terrain label assigned to each waypoint j from the measured slope and roughness of the traversed surface as follows:

$$t^j_{\text{class}} = \begin{cases} 3 & \text{if } s_j > 15^\circ \quad \text{(steep)} \\ 2 & \text{else if } r^j_{\text{imu}} > 2.0 \quad \text{(non-traversable)} \\ 1 & \text{else if } r^j_{\text{imu}} > 0.5 \quad \text{(uneven but navigable)} \\ 0 & \text{otherwise} \quad \text{(flat and fully traversable)} \end{cases} \tag{9}$$

where s_j and r^j_{imu} are the per-waypoint slope and roughness index defined in equation 10-equation 11. We ground the labels in traversal evidence to maintain a consistent taxonomy across RELLIS-3D (Jiang et al., 2021) and TartanDrive (Triest et al., 2022).

Critically, at inference time, the model does not require separate terrain attribute estimators. Instead, terrain awareness emerges implicitly through the action token selection process (as discussed in equation 14). This means that each action token in the codebook $\mathbb{A}$ inherently represents a specific combination of motion and terrain properties. When the Offroad-VLA selects an action token during inference based on camera observations and ego state, it implicitly selects both a kinematic trajectory and its associated terrain condition. Errors in token selection naturally result in suboptimal terrain choices. Therefore, we employ PRL with terrain-aware rewards (detailed in Section 3.5) to improve the VLM's ability to select terrain-safe tokens from visual inputs alone.

Table 6: Calibration of $\alpha_{t_{\text{class}}}$ over 6,000 training waypoints per dataset. **Clipped** is the fraction of waypoints reaching the cap τ=5, where all rough, fast segments become indistinguishable. $\bar{c}_v$ is the mean cost. The last column reports mean cost per traversability class. These are calibration diagnostics rather than quality metrics: neither extreme is desirable, and the target is an intermediate regime with $< 3\%$ saturation and a monotone class separation.

Scheme	RELLIS-3D (Jiang et al., 2021)			TartanDrive (Triest et al., 2022)		
	Clipped (%) ↓	$\bar{c}_v$	by class 0/1/2/3	Clipped (%) ↓	$\bar{c}_v$	by class 0/1/2/3
$\alpha = 0$ (no cost)	0.00	0.000	0.00 / 0.00 / 0.00 / 0.00	0.00	0.000	0.00 / 0.00 / 0.00 / 0.00
$0.1\times$ ours	0.00	0.041	0.00 / 0.01 / 0.19 / 0.00	0.00	0.124	0.01 / 0.07 / 0.60 / 0.01
Uniform $\alpha = 0.10$	0.00	0.916	0.03 / 0.28 / 4.05 / 0.00	24.22	46.139	1.75 / 42.55 / 140.50 / 0.20
Uniform $\alpha = 0.25$	2.33	6.126	0.08 / 0.92 / 29.85 / 0.00	50.10	85.422	19.13 / 97.23 / 146.87 / 0.72
$10\times$ ours	18.67	29.435	0.07 / 3.31 / 146.94 / 0.00	44.92	74.391	1.75 / 85.97 / 146.87 / 3.23
Ours	**0.67**	2.300	0.01 / 0.13 / 11.85 / 0.00	**2.80**	8.082	0.08 / 1.16 / 54.44 / 0.09

Slope magnitude: The second dimension captures the *slope magnitude* $s_{\text{deg}} \in \mathbb{R}$ the mean inclination in degrees of the traversed path. It is computed from the elevation profile of the ego pose trajectory of (Jiang et al., 2021; Triest et al., 2022): for each waypoint j we take the elevation change Δz_j and the horizontal displacement Δd_j with respect to the previous waypoint, and average the resulting inclination over the segment as,

$$s_{\text{deg}} = \frac{1}{N_w} \sum_{j=1}^{N_w} \arctan\left(\frac{|\Delta z_j|}{\Delta d_j} \right) \cdot \frac{180}{\pi}, \qquad \Delta d_j = \|\mathbf{p}_j - \mathbf{p}_{j-1}\|_2 \tag{10}$$

where $N_w = 10$ is the number of waypoints in the planning horizon and $\mathbf{p}_j$ is the waypoint position in the ground plane. Waypoints with $\Delta d_j \leq 0.01$ m contribute zero inclination, so that a stationary vehicle does not produce spurious slope estimates from pose noise. As the measurement is taken along the direction of travel, it is invariant to the robot's heading convention and requires no elevation map at inference time.

Roughness Index: The third dimension is the *roughness index* $r_{\text{imu}} \in \mathbb{R}$ a scalar measure of terrain irregularity derived from onboard inertial measurements. Following (Castro et al., 2023; Triest et al., 2022), it is computed as the variance of the IMU z-axis acceleration recorded while the vehicle traverses the segment, capturing the high-frequency vertical disturbances induced by surface undulations as,

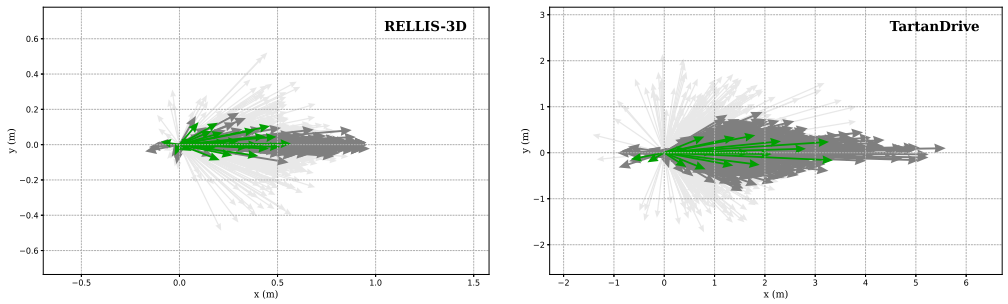
$$r^j_{\text{imu}} = \text{Var}\Big(\{a^z_k\}_{k \in \mathcal{S}_j} \Big) = \frac{1}{|\mathcal{S}_j|} \sum_{k \in \mathcal{S}_j} \left(a^z_k - \bar{a}^z_j \right)^2 \tag{11}$$

where a_k^z denotes the vertical acceleration at IMU timestep k, $\mathcal{S}_j$ is the set of IMU samples between waypoints $j-1$ and j, and $\bar{a}_j^z$ is their mean. Thus, $|\mathcal{S}_j|$ depends on the IMU sampling rate of each platform (Jiang et al., 2021; Triest et al., 2022). Since the IMU samples are uniformly spaced, no resampling is required. Retaining one value per waypoint enables the per-waypoint traversability classification in equation 9.

Velocity-Conditioned Traversability Cost: We define the fourth attribute as the *velocity-conditioned traversability cost* $c_v \in \mathbb{R}$, which captures how terrain roughness and vehicle speed jointly affect traversal difficulty. Since traversability depends on speed (Zhao et al., 2026), a surface that is safe at low speed may become hazardous at higher speed. Therefore, we compute the cost at each waypoint and average it over the planning horizon as,

$$c_v = \frac{1}{N_w} \sum_{j=1}^{N_w} \left[\exp\Big(\min\Big(\alpha_{t^j_{\text{class}}} \cdot r^j_{\text{imu}} \cdot v_j^2,\ \tau \Big) \Big) - 1 \right], \tag{12}$$

where r^j_{imu} is the local roughness index equation 11, and $v_j = \|\mathbf{p}_j - \mathbf{p}_{j-1}\|_2 / \Delta t$ is the speed at the j-th step, with $\Delta t = 0.5\,\text{s}$. The exponential term increases the penalty for high-speed traversal of rough terrain, while the quadratic dependence on speed reflects its increasing physical impact. The class-dependent coefficient $\alpha_{t^j_{\text{class}}} > 0$ further adjusts the penalty according to terrain type. We set the cap τ and the class-dependent coefficients through calibration.



(a) RELLIS-3D (b) TartanDrive

Figure 4: Visualization of the action codebook in kinematic space. Light grey curves are raw 0.5 s trajectory segments from the training split. Green marks kinematically distinct centroids. The split illustrates how the codebook allocates capacity between distinct manoeuvres and terrain-conditioned variants of the same manoeuvre.

Choice of $\alpha_{t_{\text{class}}}$: Since c_v is one of four attributes in the composite distance equation 14, we calibrate α to keep the cost informative without vanishing or saturating at the cap. Table 6 evaluates this behavior over 6,000 training waypoints per platform by scaling our chosen constants, $\{0.02, 0.05, 0.15, 0.25\}$ for RELLIS-3D and $\{0.01, 0.02, 0.03, 0.05\}$ for TartanDrive, by $0.1\times$ and $10\times$. At $0.1\times$, the cost becomes nearly constant, with a mean of 0.041 on RELLIS-3D, making its contribution to clustering negligible. At $10\times$, 18.67% of RELLIS-3D and 44.92% of TartanDrive waypoints reach the cap $\tau = 5$, making rough, high-speed segments indistinguishable. Our chosen constants avoid both extremes, with only 0.67% and 2.80% of waypoints clipped while maintaining monotonic class separation, with mean costs of 0.01, 0.13, and 11.85 for classes 0-2 on RELLIS-3D. A shared α is also unsuitable: uniform $\alpha = 0.25$ clips 2.33% of RELLIS-3D waypoints but 50.10% of TartanDrive waypoints, reflecting its higher mean speed and the quadratic dependence on velocity.

Together, these four attributes provide a compact and expressive representation of terrain conditions spanning geometric, dynamic, and semantic modalities. The full token representation for each trajectory concatenates the motion primitives with the terrain attribute vector $\boldsymbol{\phi}_j$

$$\mathbf{m}_j = \Big(\Delta x_j,\ \Delta y_j,\ \Delta\theta_j,\ t^j_{\text{class}},\ s_{\text{deg}},\ r^j_{\text{imu}},\ c_v \Big) \tag{13}$$

where Δx_j Δy_j and $\Delta\theta_j$ denote the longitudinal displacement, lateral displacement, and heading change of the j-th segment, and the remaining four components are the terrain attributes.

A.2 K-DISK CLUSTERING WITH TERRAIN AUGMENTATION

Given a corpus of N motion primitives $\{\mathbf{m}_j\}_{j=1}^{N}$, we construct the terrain-aware physical action codebook $\mathbb{A} = \{\mathbf{m}^{(1)}, \ldots, \mathbf{m}^{(K)}\}$ of K using an extended K-Disk clustering procedure (Zhang et al., 2025; Wu et al., 2024). Therefore, we extend standard K-Disk clustering to operate in the joint space using a composite distance metric as,

$$d_{m,\boldsymbol{\phi}} = \lambda_{\text{kin}} \cdot d_{\text{kin}}(\mathbf{m}_j, \mathbf{m}_k) + \lambda_{\phi} \cdot d_{\text{terrain}}(\boldsymbol{\phi}_j, \boldsymbol{\phi}_k), \tag{14}$$

where the kinematic distance is the average contour distance between trajectory boundary points,

$$d_{\text{kin}}(\mathbf{m}_j, \mathbf{m}_k) = \frac{1}{N_c} \sum_{n=1}^{N_c} \left\| \mathbf{q}_n^j - \mathbf{q}_n^k \right\|_2, \qquad N_c = 4.$$

Here $\mathbf{q}_n^j \in \mathbb{R}^2$ denotes the n-th corner of the vehicle footprint placed at the end pose $(\Delta x_j, \Delta y_j, \Delta\theta_j)$ of primitive j. The footprint is a rectangle and its corners are enumerated in a fixed order (such as, left-front, right-front, right-back, left-back). Therefore $\mathbf{q}_n^j$ and $\mathbf{q}_n^k$ correspond to the same physical corner. Instead of center positions, comparing footprint corners makes the distance sensitive to heading differences. Moreover, two primitives ending at the same point with different orientations are correctly separated. The terrain term is the normalized attribute distance and represented as

$$d_{\text{terrain}}(\boldsymbol{\phi}_j, \boldsymbol{\phi}_k) = \frac{1}{d_\phi} \left\| \frac{\boldsymbol{\phi}_j - \boldsymbol{\phi}_k}{\boldsymbol{\sigma}_\phi} \right\|_2, \qquad d_\phi = 4,$$

Table 7: Unified versus per-platform codebooks with $K = 2048$.

	RELLIS-3D		TartanDrive	
Codebook	ADE ↓	CC ↑	ADE ↓	CC ↑
Unified, pooled	0.0328	12.7	0.0324	99.3
Unified, speed-normalized	0.0200	46.7	0.0336	99.0
Unified, speed-normalized + balanced	0.0163	49.5	0.0345	98.8
Per-platform (ours)	**0.0051**	**83.1**	**0.0322**	**99.8**

where $\boldsymbol{\sigma}_\phi$ is the per-dimension standard deviation across the full segment corpus, providing consistent scaling across terrain attribute dimensions with different physical units. The hyperparam λ_{kin} and λ_ϕ (with $\lambda_{kin} + \lambda_\phi = 1$) are selected using grid search over $\lambda_{kin} \in \{0.3, 0.5, 0.7, 0.9\}$ on a held-out validation split, optimizing jointly for terrain-stratified reconstruction ADE and codebook coverage. We find $\lambda_{kin} = 0.7$ and $\lambda_\phi = 0.3$ to be optimal, reflecting that kinematic diversity is the primary driver of codebook expressiveness while terrain diversity serves as a regularizing signal (see Table 5b).

A.3 CODEBOOK EVALUATION

We evaluate codebook quality through trajectory reconstruction. For each ground-truth trajectory, we follow the same tokenization procedure used during training (assigning one token to every 0.5s step). Each token then placed at the previously reconstructed pose and decode the sequence autoregressively, allowing quantization errors to accumulate throughout the rollout. We evaluate all tokenizers using this same per-step rollout procedure.

A.3.1 WHY PER-PLATFORM CODEBOOKS?

VLA-based on-road navigation systems typically use a single action codebook, e.g., one fitted on the Waymo Open Motion Dataset (WOMD) (Ettinger et al., 2021). This works well on-road, where all vehicles operate within similar speed regimes. Off-road settings, however, differ substantially in their platforms. RELLIS-3D uses a small ground robot, whereas TartanDrive uses an all-terrain vehicle. Figure 4 shows how the codebook adapts to the motion characteristics of each platform. RELLIS-3D operates at 1.18 m/s and covers 5.96 m per horizon, giving a compact action distribution whose lateral-to-forward extent ratio is larger (0.31 versus 0.26). Its capacity is therefore spent on

resolving heading rather than distance. In contrast, TartanDrive operates at 3.26 m/s and covers 16.38 m per horizon, producing a longitudinal spread roughly six times wider that instead requires finer resolution of forward motion.

These differences motivate fitting the codebook separately for each dataset, which we validate experimentally. We first test whether a single codebook can accommodate both platforms (as demonstrated in Table 7). Directly pooling the two corpora substantially degrades RELLIS-3D performance. TartanDrive contributes approximately $9\times$ more primitives and spans a wider metric range, so RELLIS-3D retains only 12.7% of the vocabulary while its reconstruction error increases by $6.4\times$. We then normalize each primitive's translation by the reference step length of its platform, allowing the codebook to cluster dimensionless manoeuvre shapes that are rescaled to the appropriate platform during tokenization. This improves RELLIS-3D coverage and reconstruction, and balancing the two corpora gives a further improvement. However, a substantial gap remains.

Table 8: Vocabulary size vs. reconstruction and planning. * represents deployed configuration.

Platform	K	Sample Per Token	Reconstruction ADE ↓	CC ↑	ADE-5s ↓	FDE-5s ↓	Truncated ↓
RELLIS-3D	256*	90	0.0265	**96.5**	**0.4180**	**0.8145**	**0 / 380**
	1024	22	0.0106	88.6	0.5285	1.0022	**0 / 380**
	2048	13	0.0051	83.1	0.5284	1.0231	97 / 380
	4096	5	**0.0012**	60.5	0.7444	1.6314	**0 / 380**
TartanDrive	1024*	150	0.0428	**100.0**	**2.4562**	**5.4365**	**0 / 1803**
	2048	98	0.0322	99.8	2.5388	5.6880	**0 / 1803**
	4096	38	**0.0239**	99.5	2.5759	5.8380	22 / 1803

Even after normalization and balancing, RELLIS-3D coverage plateaus at about 49%, leaving roughly half of the shared vocabulary assigned to manoeuvres that do not occur on this platform. Scale-free manoeuvre statistics explain why. Heading change per unit distance is similar across the two platforms (median $|\Delta\theta|$ of 0.0169 versus 0.0191 rad/m), but lateral displacement relative to step length differs substantially (p90 of 0.119 for RELLIS-3D versus 0.295 for TartanDrive). The all-terrain vehicle thus exhibits roughly $2.5\times$ greater lateral slip per unit displacement than the ground robot. This difference in vehicle dynamics cannot be removed by metric rescaling, which motivates us per-platform codebook constructions.

A.3.2 Vocabulary Size Selection

Table 8 reports the quantization ceiling for each vocabulary size, together with the end-to-end planning results for the sizes used to train a model. Reconstruction error decreases monotonically with K across the platforms (Jiang et al., 2021; Triest et al., 2022). Therefore, reconstruction error alone would favor the largest vocabulary. However, reconstruction is not the limiting factor, as RELLIS-3D with $K = 256$ achieves a reconstruction error of only 0.027m. Instead, increasing K reduces the number of training examples available for each token, making token prediction more difficult. This trend is consistent across platforms. As observered from the Table 8, larger vocabularies improve reconstruction but can degrade planning performance. Particularly, on RELLIS-3D, $K = 2048$ provides only 13 training examples per token. Therefore, it causes 97 of 380 test trajectories to terminate before producing the required ten action tokens, and increases ADE-5s from 0.418 to 0.528 m. Similarly, TartanDrive shows the same behavior at $K = 4096$ where each token receives 38 training examples, 22 of 1803 trajectories are truncated, and ADE-5s increases from 2.456 to 2.576m. Thus, larger vocabularies can reduce reconstruction error while making trajectory prediction less reliable, primarily because the model fails to generate complete trajectories.

Table 9: Contribution of K-Disk extensions.

Platform	K	Greedy K-Disk	**+ all three**	Δ
RELLIS-3D	256	0.0301	**0.0265**	-12%
TartanDrive	1024	0.0586	**0.0428**	-2696%

A.3.3 Contribution of the K-Disk extension

The plain greedy K-Disk covers the entire corpus but does not minimize quantization error. It places each center at a randomly selected primitive and never revisits that choice. We therefore extend the greedy procedure with three components:

1. **Farthest-point seeding.** We place each new centre at the primitive currently worst served by the existing centres, rather than selecting centres uniformly at random. This improves coverage of sparse regions that uniform seeding tends to miss.
2. **Budget filling.** The greedy procedure stops once all primitives are covered, which can leave the codebook short of K when the budget is large.
3. **Lloyd refinement.** We perform ten alternating passes that assign each primitive to its nearest centre under equation 14 and update each centre using the mean of its assigned primitives, with headings averaged on the unit circle. At $K = 256$ on RELLIS-3D this reduces the mean quantization distance from 0.0718 to 0.0491, with the centroid shift falling from 0.023 to 0.0005, indicating convergence.

Table 9 evaluates the contribution of each of the three components for the deployed vocabulary size. They reduce reconstruction error on every platform, by 12% on RELLIS-3D, and 26% on TartanDrive, and account for essentially all of the gap between plain greedy K-Disk and the geometric tokenizers.

A.3.4 Token Integration into the Language Model

The terrain-aware physical action tokens are incorporated into the VLM vocabulary as additional special tokens $\langle \texttt{action}_0 \rangle, \langle \texttt{action}_1 \rangle, \ldots, \langle \texttt{action}_{K-1} \rangle$. During training, continuous trajectory segments are discretized by mapping each 0.5-second segment to its nearest token in $\mathbb{A}$ using equation 14. During inference, the model autoregressively generates a sequence of $T = 10$ action tokens, each representing 0.5 s of motion, yielding a 5-second planning horizon. The spatial resolution of each token is implicitly speed-dependent. This speed dependent spatial structure is naturally captured by the kinematic component $(\Delta x_k, \Delta y_k, \Delta \theta_k)$ of each token, which encodes the actual displacement rather than a fixed spatial step.

B Affordance-Centric Chain-of-Thought Reasoning

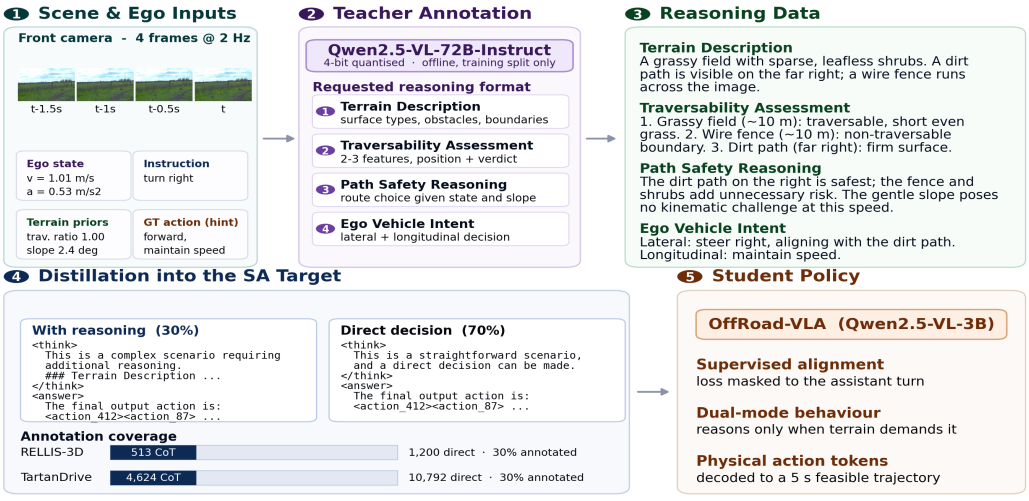


Figure 5: Automated reasoning annotation and distillation pipeline for off-road driving.

B.1 Reasoning Schema

Prompt Design: The system prompt defines the model as an autonomous off-road navigation system and specifies the required output format using `<think>` and `<answer>` tags (as showin in Fig. 5). We provide three representative off-road reasoning examples in the user prompt and structure the reasoning into four steps: ① *Terrain Description*, identifying visible surfaces and boundaries. ② *Traversability Assessment*, evaluating terrain properties and obstacles. ③ *Path Safety Reasoning*, relating terrain conditions to vehicle states such as slope and velocity. and ④ *Ego Vehicle Intent*, selecting the final lateral and longitudinal action.

Input Conditioning: The user message provides front-view camera images, ego-vehicle states, and off-road driving instructions. We additionally provide the ground-truth driving meta-action derived from RELLIS-3D (Jiang et al., 2021) and TartanDrive (Triest et al., 2022) trajectory as an explicit hint. This guides the model to generate causal reasoning that connects terrain conditions and vehicle state to the selected action, reducing irrelevant or inconsistent reasoning.

Reasoning Generation: We use Qwen2.5-VL-72B (Bai et al., 2025) as the reasoning annotation model to generate terrain-grounded reasoning for unstructured off-road scenes, as illustrated in Fig. 5. We limit the generated reasoning to a maximum of 700 tokens.

This proposed pipeline reduces reliance on manual annotation and enables knowledge distillation from a large teacher model to a compact target model.



(a) RELLIS-3D (Jiang et al., 2021) (b) TartanDrive (Triest et al., 2022)

Figure 6: Visualization of the affordance-centric COT reasoning annotation on off-road datasets.

B.2 Sanity Check

To verify the quality of the generated reasoning annotations, human annotators evaluated 1,000 randomly sampled instances from both dataset RELLIS-3D and TartanDrive, using a binary scoring scheme: a sample receives a score of 1 only if terrain object identification, causal reasoning, and

action decisions are jointly accurate and complete, and 0 otherwise. This strict all-in-one criterion gives an average annotation accuracy of 83.47%, confirming the reliability of the proposed pipeline. Samples failing this criterion are discarded from the training set.

B.3 Affordance-Centric COT Reasoning Annotation Visualization

Fig. 6 presents CoT reasoning examples from RELLIS-3D and TartanDrive, demonstrating that the annotation pipeline captures terrain-specific cues such as slope, surface conditions, vegetation, and traversability. The annotations further connect these observations to appropriate driving actions, such as reducing speed, changing direction, or selecting a safer path, thereby grounding the reasoning in terrain-aware action selection.

C Supervised Adaptation

Off-road navigation presents unique training challenges due to terrain diversity, class imbalance (flat terrain dominates while hazardous classes are rare), and the need for multimodal sensor fusion across varied lighting and weather conditions. We perform SA using PyTorch Lightning with gradient checkpointing and bfloat16 mixed precision for memory-efficient single-GPU training on each datasets. Our pipeline uses Fully Sharded Data Parallel (FSDP) (Zhao et al., 2023) with automatic transformer-layer wrapping. In our single-GPU setting, FSDP incurs no cross-device sharding overhead, and together with gradient checkpointing (Chen et al., 2016) and bfloat16 mixed precision enables full-parameter training on a single GPU (Ren et al., 2021). Mixed-precision training with BFloat16 for param, gradients, and buffers reduces memory usage while maintaining numerical stability across diverse terrain representations. Gradient checkpointing is enabled in the VLM backbone to accommodate high-resolution outdoor imagery necessary for terrain perception. Fig. 7 shows that the **System Prompt** defines the model's role as an off-road navigation system, specifying the affordance-centric CoT reasoning format that grounds decisions in terrain properties. Moreover, the **User Message** includes image sequences from the forward-facing camera, ego vehicle state, and high-level navigation instructions.

System Prompt

- **Role:**
  You are an autonomous off-road navigation system operating on unstructured terrain.
- **Task:**
  You will receive front-view camera observations and the vehicle's current kinematic state. Your task is to predict a safe and kinematically feasible driving action for the next five seconds.
- **Fast/Slow Thinking:**
  Carefully analyse terrain traversability. If the terrain is complex or hazardous, use step-by-step reasoning (Chain-of-Thought) to arrive at the best driving action. Otherwise, you may directly predict the final action.

  Structure your reasoning as follows:
  1. **Terrain Description:** Describe the terrain ahead, identifying surface types (gravel, mud, grass, rock) and boundaries.
  2. **Traversability Assessment:** Identify two or three key terrain features or obstacles and their positions relative to the ego vehicle.
  3. **Path Safety Reasoning:** Reason about the safest path given slope, surface roughness, wheel-slip risk and kinematic feasibility.
  4. **Ego Vehicle Intent:** Based on the observed terrain and current vehicle state, reason about the desired intent of the ego vehicle.
  5. **Final Action Decision:** Select one lateral action and one longitudinal action:
     - **Lateral actions** (choose exactly one): [move forward, steer left, steer right, sharp left, sharp right]
     - **Longitudinal actions** (choose exactly one): [stop, decelerate to zero, maintain speed, slow deceleration, deceleration, slow acceleration, acceleration]

  Present the final action clearly after your reasoning steps.

User Message

- **Input Description:**
  The vehicle is equipped with a forward-facing camera and navigates unstructured off-road terrain using on-board perception alone.
- **Camera Streams:**
  - **Front Camera:**
    The video presents the forward view of the vehicle, comprising four sequential frames sampled at 2 Hz.
    [front camera frames]
- **Terrain Priors:**
  Measured traversability ratio is **85%,** and the estimated slope ahead is **5.2 deg.**
- **Ego States:**
  The current velocity of the vehicle is **1.2 m/s,** and the current acceleration is **0.67 m/s2.**
- **Instruction:**
  The navigation goal is **Navigate to Goal.** Based on this information, plan a safe off-road action trajectory for the vehicle over the next five seconds.

Figure 7: **System Prompt** and **User Message** is used for supervised adaption.

D Safety-Aware Post-Trained using Reinforcement Learning

GRPO employs group-based sampling to compute the advantage function, replacing conventional state-value estimators or critic models. This design accelerates training while aligning naturally with the inherent multimodality of planning, which requires evaluating and selecting from a set of candidate trajectories. The overall PRL procedure is illustrated in Algorithm 1.

Algorithm 1 Safety-Aware Post-Training of OffRoad-VLA with GRPO

Input: π_{SA}, $\mathcal{G}$, K, $\mathcal{A}$, $\{\lambda_p, \lambda_a, \lambda_s, \lambda_t, \lambda_{sl}, \lambda_c, \lambda_r\}$, ξ, ε
Output: π_{PRL}

```
Initialize π_ref ← π_SA, π_θ ← π_SA, π_θold ← π_SA
for k = 1, ..., K do
    Sample query q and G outputs 𝕆 = {o_i}_{i=1}^G ~ π_θold(·|q)
    for i = 1, ..., G do
        Decode τ_i ← A(o_i) and compute r_i using equation 15
        Compute ρ_i, A_i, and D_KL^(i) using equation 7-equation 8
    Compute 𝕁_G(θ) using equation 5
    Update θ ← arg max_θ 𝕁_G(θ)
    π_θold ← π_θ
return π_PRL ← π_θ
```

D.1 Off-Road Reward Design

The off-road reward function is the core of the PRL stage. We define a composite reward r as,

$$r = \lambda_p r_{\mathrm{pro}} + \lambda_a r_{\mathrm{acc}} + \lambda_s r_{\mathrm{smooth}} - \lambda_t r_{\mathrm{trav}} - \lambda_{sl} r_{\mathrm{slope}} - \lambda_c r_{\mathrm{col}} - \lambda_r r_{\mathrm{CoT}} \tag{15}$$

The progress reward measures net displacement from the starting pose. Therefore oscillatory and looping trajectories receive no credit for distance traveled without advancement. Moreover, it is defined as,

$$r_{\mathrm{pro}} = \min\left(\frac{\|\mathbf{p}_T - \mathbf{p}_0\|}{d_{\mathrm{target}}}, r_{\max}\right)$$

where $r_{\max}$ is the maximum progress reward, and d_{target} is dataset-specific five-second displacement.

The accuracy reward provides a dense imitation signal toward the expert trajectory as,

$$r_{\mathrm{accuracy}} = \exp\left(-\frac{\mathrm{ADE}}{\tau}\right), \qquad \mathrm{ADE} = \frac{1}{T}\sum_{t=1}^{T}\|\hat{\mathbf{P}}_j - \mathbf{P}_j\|$$

with $\mathbf{p}_j$ the ground-truth waypoint and $\tau = 1.0$ m. Unlike the corridor-based collision penalty, this term is non-flat everywhere and therefore supplies gradient even for trajectories already inside the expert corridor.

The smoothness reward penalizes high jerk and abrupt heading changes

$$r_{\mathrm{smooth}} = \exp\left(-\frac{\bar{j} + \lambda_h \bar{\dot{\theta}}}{\sigma}\right)$$

where $\bar{j}$ is the mean magnitude of the third finite difference of position, $\bar{\dot{\theta}}$ the mean absolute heading change.

The traversability and slope penalties are computed purely at the token level, as the fraction of emitted action tokens carrying unsafe terrain codes as follows:

$$r_{\mathrm{trav}} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{1}[c(a_t) \in \{2,3\}], \qquad r_{\mathrm{slope}} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{1}[c(a_t) = 3]$$

where $c(a_t)$ is the terrain code assigned to action token a_t during codebook construction from ground-truth panoptic and LiDAR-derived elevation annotations (RELLIS-3D (Jiang et al., 2021), and TartanDrive (Triest et al., 2022)). We exclude scene-level terrain statistics from the reward: such terms are constant with respect to the policy for a given scene and therefore contribute variance to the advantage without supplying learning signal. At inference time the model operates on camera observations alone, having learned during PRL to avoid unsafe action tokens from visual features.

The collision penalty measures deviation beyond a safe corridor around the expert path as

$$r_{\mathrm{col}} = \frac{1}{T}\sum_{t=1}^{T}\min\left(1,\ \max\left(0,\ \frac{d_{\mathrm{gt}}(t) - r_{\mathrm{corridor}}}{r_{\mathrm{corridor}}}\right)\right)$$

serving as a conservative surrogate for physical obstacle interaction in the absence of closed-loop environment feedback.

The CoT efficiency penalty discourages unnecessarily long reasoning chains:

$$r_{\text{CoT}} = \begin{cases} \sigma((L - L_c)\gamma) & \text{if "}\texttt{complex scenario}\text{" in text} \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

where L is the number of generated tokens, $L_c = 400$ and $\gamma = 0.002$. This steers the model DAT in simple terrain while preserving AT in complex scenarios.

Table 10: Dataset specific parameters for PRL.

Parameters	**RELLIS-3D**	**TartanDrive**
d_{target}	5.0	15.0
$r_{\max}$	1.25	1.25
λ_h	0.2	0.2
σ	0.45	0.5
r_{corridor}	1.2	1.5

E Experimental Setup

E.1 Evaluation Benchmarks and Metrics

We organize evaluation metrics into two tiers. First, codebook quality is assessed through average displacement error (ADE) (Caesar et al., 2020), final displacement error (FDE) (Phong et al., 2023), and codebook coverage (CC). ADE measures the L2 distance between ground-truth and reconstructed trajectory waypoints, while FDE measures the endpoint error at the planning horizon. CC quantifies the percentage of ground-truth trajectory segments in the test set that can be successfully represented by valid token sequences from the codebook. Mainly, it verify that the codebook uniformly covers diverse terrain maneuvers, including hazardous classes. Second, we employ terrain-specific metrics for planning quality, safety, and adaptability across multiple evaluation settings.

In-Domain Open-Loop Evaluation: We evaluate held-out test sets of RELLIS-3D (Jiang et al., 2021) and TartanDrive (Triest et al., 2022). Metrics are organized into two ways:

- *Planning Quality:* We evaluate trajectory accuracy using ADE at 1 s, 3 s, and 5 s, along with FDE at 5 s (Caesar et al., 2020). These metrics measure prediction error at multiple time horizons. Beyond point-wise accuracy, we assess long-horizon planning consistency using terrain-conditioned route completion (TCRC) (Karnan et al., 2023; Lee et al., 2023), the fraction of scenarios in which the predicted $5\,\text{s}$ trajectory reaches the goal within $1\,\text{m}$ while every waypoint remains on traversable terrain. For safety evaluation, we report collision rate (Li et al., 2024), which measures intersection with non-traversable regions. We also compute traversability violation rate (TVR) (Hagmanns et al., 2025), which penalizes waypoints in terrain classes 2 (non-traversable) or 3 (steep). This captures terrain-unsafe planning beyond physical collisions.
- *Terrain-Awareness:* We measure discrete planning accuracy using token prediction accuracy (TPA) at Top-1 and Top-5 (Pertsch et al., 2025). This metric assesses whether the model selects the correct action token from the codebook. Slope error (SE) measured by mean absolute error (MAE) (Frey et al., 2024; Sivaprakasam et al., 2024) evaluates how accurately the model captures terrain inclination along the planned trajectory. Roughness correlation (RC) (Castro et al., 2023; Seo et al., 2024) assesses the model's ability to predict surface characteristics. These physical terrain estimates are then aggregated into a single safety score (SS), computed as $1 - \text{TVR}_{\text{weighted}}$. This score weights traversability violations by their severity to produce a comprehensive terrain-safety summary.

Cross-Dataset Generalization: For cross-dataset evaluation, we report both *Planning Quality* and *Terrain-Awareness* metrics to assess whether terrain-awareness generalizes beyond trained terrain patterns. Additionally, we report miss rate (MR_τ), defined as the percentage of samples whose final displacement error exceeds a threshold τ evaluated at $\tau \in \{2, 4\}$ m to capture failure frequency at both strict and relaxed tolerance levels.

Closed-Loop Simulation: Open-loop evaluation measures trajectory prediction quality, it does not capture whether those predictions translate into safe, executable control when deployed on a vehicle. Closed-loop simulation addresses this gap by testing the model in a feedback control setting where predictions are executed and the vehicle state continuously affects future planning decisions. We deploy OffRoad-VLA in the MAVS physics-based simulator (Hudson et al., 2020; Goodin et al., 2022; Carruth et al., 2024), which provides realistic vehicle-terrain interaction for off-road environments. The simulator operates at 2 Hz replanning frequency to match real-time deployment constraints. We evaluate navigation performance using four end-to-end metrics: Success Rate (SR), Driving Score (DS), average speed and TCRC. SR measures successful goal completion without critical failures, such as collisions or terrain violations, while DS combines route completion with infraction penalties, and average speed is reported in *m/s*.

E.2 Implementation Details

We implement OffRoad-VLA using a standard VLA architecture adapted for off-road navigation. The model receives input from a single front-facing camera with four history frames sampled at 2 Hz. It predicts a 5-second trajectory represented by 10 future poses sampled at 0.5-second intervals, as illustrated in Fig. 6.

For the SA baseline, we fine-tune Qwen2.5-VL-3B (Bai et al., 2025) with gradient checkpointing enabled. The learning rate is initialized to 2.0×10^{-5} with a 500-step warm-up to ensure stable learning across imbalanced terrain classes, followed by a 2% decay every 2,000 steps. Training proceeds for 5 epochs with an effective batch size of 4 (batch size 1 with gradient accumulation over 4 steps) on a single NVIDIA 96GB Blackwell GPU. Gradient clipping at 1.0 prevents instability from high-gradient terrain transitions (e.g., sudden slope changes or traversability boundaries). We set $\lambda_{\text{cot}} = 40$, $\lambda_t = 2.0$, and $\lambda_a = 1.0$, selected through grid search. Sensitivity analysis showed less than 3% variation in trajectory violation rate across $\lambda_t \in \{1.0, 1.5, 2.0\}$ confirming robustness to the exact choice of weighting values.

For PRL, we initialize from the best supervised checkpoint and train LoRA adapters to preserve the base model. The LoRA configuration uses rank $r = 8$ scaling factor $\alpha = 8$ and dropout rate of 0.1 to balance expressiveness and regularization. The GRPO training procedure operates with batch size 1, learning rate 2×10^{-5} and runs for 1 epoch. Rewards are scaled by $\kappa = 10$ and converted to advantages by group normalization over $\mathcal{G} = 4$ completions sampled per scene, with a KL penalty of $\xi = 0.04$ against the frozen SA reference policy. We set the reward hyperparameters as $\lambda_p = 1.4$, $\lambda_a = 1.5$, $\lambda_s = 0.35$, $\lambda_t = 1.2$, $\lambda_{sl} = 0.9$, $\lambda_c = 1.1$, and $\lambda_r = 0.3$. The others dataset specific hyperparameter's are discussed in Table 10.

E.3 Model Training:

We use the same data collection and input configuration for RELLIS-3D and TartanDrive, while training separate OffRoad-VLA models for each dataset to capture their distinct vehicle dynamics and terrain characteristics. Both models initialize from the same Qwen2.5-VL-3B-Instruct backbone and use the same prompt template, and supervised training objective. The visual input consists of a single front-facing camera, with four frames sampled at 2 Hz to provide a 1.5 s temporal history. Each frame is encoded using a fixed budget of 140 visual tokens ($28 \times 28 \times 140$ pixels). The ego-state input includes the current vehicle velocity and acceleration, along with terrain priors consisting of the measured traversability ratio and estimated slope, and a high-level navigation instruction. For closed-loop evaluation, we additionally train a separate model on MAVS rollouts using the same input configuration.

E.4 INFERENCE:

At evaluation time, we use greedy decoding with a 2048-token budget for both reasoning and action generation. The action sequence consists of 10 tokens from the terrain-aware K-Disk vocabulary, which are decoded into the corresponding waypoints. The reasoning mode is determined by the trained policy rather than the decoding strategy, as the training data contains both chain-of-thought and direct-action targets. Thus, the model learns to select the appropriate reasoning depth based on scene complexity, while greedy decoding ensures deterministic and reproducible evaluation. During PRL, we use stochastic sampling with temperature 0.9, top-p=1.0, and no top-k filtering. Each completion is limited to 320 new tokens, providing sufficient diversity among the $\mathcal{G}$ samples for reliable advantage estimation.



Direct action generation ADE-5s 0.154 m FDE-5s 0.465 m

Thinking:
This is a straightforward scenario, no need of additional reasoning.

Action:
<offroad_action_1080> <offroad_action_380> <offroad_action_1011> <offroad_action_1308> <offroad_action_1080>

<offroad_action_1080> <offroad_action_1080> <offroad_action_1080> <offroad_action_1080> <offroad_action_1080>

(a) Direct action generation



Adaptive thinking ADE-5s 3.468 m FDE-5s 6.101 m

Thinking:
This is a complex scenario requiring additional reasoning.

Terrain Description
The terrain ahead consists primarily of a grassy field with some patches of bare soil. There is a white object (possibly a piece of machinery or equipment) located near the center-right of the image, slightly elevated compared to the surrounding grass. The horizon is lined with sparse trees and bushes, indicating a relatively open area. The sky is clear with scattered clouds, suggesting fair weather conditions. There are no visible water bodies, rocks, or significant elevation changes apart from the slight elevation of the white object.

Traversability Assessment
1. Grassy Field (center-left, immediate foreground):
- Traversability: Traversable. The grass appears short and even, providing good traction for the vehicle.
- Position: Center-left, immediately ahead of the vehicle.
2. White Object (center-right, ~5-10 m ahead):
- Traversability: Potentially traversable. While the object itself might pose a minor obstacle, it does not appear to be a significant barrier. However, care should be taken to avoid any hidden hazards behind it.
- Position: Center-right, approximately 5-10 meters ahead.
3. Sparse Trees and Bushes (horizon, far background):
- Traversability: Non-traversable. These form a natural boundary and should be avoided as they could damage the vehicle.
- Position: Far background, spanning across the horizon.

Path Safety Reasoning
The grassy field provides a stable and even surface for the vehicle to traverse. The slight elevation of the white object does not present a significant challenge, as it can likely be driven over without issue. The sparse trees and bushes in the background serve as a natural boundary, but they are far enough away that they do not impact the immediate path. Given the current vehicle state, maintaining a steady speed while moving forward seems appropriate, as there are no immediate obstacles or hazards in the immediate vicinity that would require a change in direction or speed.

Best Driving Action
Given the terrain analysis and the absence of immediate obstacles, the safest and most efficient course of action is to continue moving forward at a maintained speed. This aligns with the ground-truth best action provided.
- Lateral Action: move forward
- Longitudinal Action: maintain speed

This approach ensures the vehicle stays on the safest part of the terrain while avoiding unnecessary maneuvers, making it the optimal choice for the given conditions.

Action:
<offroad_action_380> <offroad_action_1080> <offroad_action_1011> <offroad_action_1080> <offroad_action_1080> <offroad_action_1080>
<offroad_action_1080> <offroad_action_1080> <offroad_action_1080> <offroad_action_1080>

(b) Adaptive thinking

Figure 8: Planning and reasoning results of Offroad-VLA (SA) on the RELLIS-3D dataset.

F OPEN-LOOP PLANNING AND SAFETY

F.1 QUALITATIVE ANALYSIS

Figure 8 visualizes SA action generation on the RELLIS-3D dataset. Figure 8a shows a simple scenario where OffRoad-VLA directly generates the action without additional reasoning, whereas Figure 8b shows a complex terrain scenario where the model uses affordance-centric reasoning to assess traversability, slope, and obstacles before selecting the appropriate action.

Similarly, Figure 9 presents SA action generation on the TartanDrive dataset. The model directly generates actions in simple terrain, while complex scenarios trigger affordance-centric reasoning that considers terrain conditions, traversability, and obstacles before producing the final action. These

examples demonstrate the model's ability to adapt its reasoning depth to the complexity of the off-road environment.

F.2 Terrain-Aware Safety Evaluation



Direct action generation ADE-5s 2.297 m FDE-5s 4.658 m

Thinking:
This is a straightforward scenario, no need of additional reasoning.

Action:
<offroad_action_3011> <offroad_action_2967> <offroad_action_3011> <offroad_action_2967> <offroad_action_2967> <offroad_action_2967>

<offroad_action_2967> <offroad_action_2967> <offroad_action_2967> <offroad_action_2967>

(a) Direct action generation



Adaptive thinking ADE-5s 0.281 m FDE-5s 0.666 m

Thinking:
This is a complex scenario requiring additional reasoning.

Terrain Description
The terrain ahead consists of a mix of surfaces including gravel, dirt, and scattered debris such as small rocks and pieces of wood. The ground appears uneven with some areas showing signs of wear from previous vehicle traffic. There is dense vegetation on both sides of the path, which acts as a natural boundary. The path seems to be a worn-out road or track, possibly abandoned, with remnants of larger stones and broken pavement pieces scattered across it.

Traversability Assessment
1. Scattered Debris (center, ~5 m ahead): Potentially traversable. The debris includes small rocks and pieces of wood which could cause minor bumps but do not pose a significant obstacle. Careful navigation is required to avoid damaging the vehicle.
2. Broken Pavement Pieces (center-right, ~10 m ahead): Non-traversable. These large pieces of broken concrete can cause damage to the vehicle if driven over directly. They should be avoided by steering slightly to the left.
3. Dense Vegetation (both sides): Non-traversable. The vegetation forms a natural barrier that restricts passage. The vehicle should stay within the confines of the existing path to avoid getting stuck or damaging the undercarriage.

Path Safety Reasoning
The safest traversable path lies along the center of the existing road, avoiding the larger debris and the dense vegetation. The surface is primarily gravel and dirt, providing reasonable traction despite the unevenness. The presence of scattered smaller rocks suggests a need for cautious driving to prevent wheel slip or damage. Given the current vehicle state, maintaining a controlled speed while navigating through the debris will minimize risks. The slight incline and unevenness require careful management of speed and steering to ensure stability and prevent loss of control.

Ego Vehicle Intent
Given the observations and the need to navigate safely through the terrain:
- Lateral Action: steer left. This will help avoid the larger debris on the right and keep the vehicle centered on the more stable part of the path.
- Longitudinal Action: slow deceleration. Reducing speed gradually will allow the vehicle to handle the uneven terrain and scattered debris more safely, minimizing the risk of wheel slip or damage.

This aligns with the ground-truth best action of move forward with slow deceleration, ensuring a safe and controlled traversal of the challenging terrain ahead.

Action:
<offroad_action_850> <offroad_action_806> <offroad_action_353> <offroad_action_386> <offroad_action_407> <offroad_action_580>
<offroad_action_908> <offroad_action_933> <offroad_action_640> <offroad_action_905>

(b) Adaptive thinking

Figure 9: Planning and reasoning results of Offroad-VLA (SA) on the TartanDrive dataset.

Table 11: Terrain-aware planning on RELLIS-3D. TPA is computed over the $K{=}256$ action codebook (random chance: 0.39% Top-1, 1.95% Top-5). The baselines do not emit action tokens and are therefore not applicable. TCA is terrain classification over four traversability classes.

Method	TPA ↑		TCA (%)↑	SE ↓ (MAE)	RC ↑	SS ↑
	Top-1	Top-5				
Terrax (Li et al., 2025b)	–	–	49.37	3.29°	-0.07	0.84
Sterling (Karnan et al., 2023)	–	–	51.63	2.42°	0.09	<u>0.91</u>
GPT-DRIVER (Mao et al., 2023a)	–	–	1.75	2.18°	-0.03	0.77
MetaVQA (Wang et al., 2025a)	–	–	**95.74**	1.08°	-0.06	**0.97**
Alpamayo (Wang et al., 2025b)	–	–	48.68	1.01°	-0.01	0.75
OffRoad-VLA (SA)	2.39	12.03	79.76	<u>0.60°</u>	**0.20**	0.85
OffRoad-VLA (PRL)	**2.74**	**13.29**	<u>84.11</u>	**0.59°**	<u>0.13</u>	0.84

RELLIS-3D Analysis: Table 11 highlights the gap between terrain perception and terrain-aware action generation. MetaVQA (Wang et al., 2025a) achieves the highest TCA of 95.74% and a strong SS of 0.97, although its RC is negative. It indicates weak alignment between the predicted trajectories and terrain properties. Similarly, other perception-oriented methods show similar limitations, with Terrax (Li et al., 2025b) and GPT-DRIVER (Mao et al., 2023a) also obtaining negative RC values. On contrary OffRoad-VLA explicitly connects terrain understanding with action generation. PRL improves TPA from 2.39% to 2.74% Top-1 and from 12.03% to 13.29% Top-5 over the SA baseline,

while TCA increases from 79.76% to 84.11%. It also slightly reduces slope error from $0.60°$ to $0.59°$. The positive correlation indicates that the generated trajectories remain aligned with terrain properties. Overall, these results show that PRL improves terrain-aware action prediction while maintaining comparable safety performance.

Table 12: Terrain-aware planning on TartanDrive. TPA is computed over the $K{=}1024$ action codebook (random chance: 0.10% Top-1, 0.49% Top-5). The baselines do not emit action tokens and are therefore not applicable. TCA is terrain classification over four traversability classes.

Method	TPA ↑		TCA (%) ↑	SE ↓ (MAE)	RC ↑	SS ↑
	Top-1	Top-5				
Terrax (Li et al., 2025b)	–	–	73.09	4.23°	-0.12	0.83
Sterling (Karnan et al., 2023)	–	–	33.31	6.14°	-0.13	0.81
GPT-DRIVER (Mao et al., 2023a)	–	–	19.04	5.64°	-0.27	<u>0.87</u>
MetaVQA (Wang et al., 2025a)	–	–	68.43	4.15°	-0.04	**0.89**
Alpamayo (Wang et al., 2025b)	–	–	85.21	1.14°	0.45	0.85
OffRoad-VLA (SA)	1.19	5.22	<u>85.62</u>	**0.99°**	<u>0.50</u>	0.85
OffRoad-VLA (PRL)	**1.92**	**6.96**	**89.85**	<u>1.00°</u>	**0.52**	0.84

TartanDrive Analysis: Table 12 evaluates terrain-aware planning under the high-speed and dynamically challenging conditions of TartanDrive. OffRoad-VLA (PRL) getting better TPA, improving Top-1 from 1.19% to 1.92% and Top-5 from 5.22% to 6.96% over SA. It also improves TCA from 85.62% to 89.85% and RC from 0.50 to 0.52, while maintaining a comparable SE of $1.00°$. Among the baselines, Alpamayo (Wang et al., 2025b) provides the closest terrain-aware performance, with TCA of 85.21%, SE of $1.14°$, and RC of 0.45, but remains below OffRoad-VLA on these metrics. These results show that PRL further improves terrain-aware action prediction and maintains strong alignment between the generated trajectories and terrain properties.



Adaptive thinking ADE-5s 2.740 m FDE-5s 7.684 m

Thinking:
This is a straightforward scenario, and a direct decision can be made.

Action:
<offroad_action_776> <offroad_action_515> <offroad_action_812> <offroad_action_838> <offroad_action_418> <offroad_action_43>

<offroad_action_16> <offroad_action_168> <offroad_action_137> <offroad_action_128>

(a) Direct action generation



Adaptive thinking ADE-5s 9.905 m FDE-5s 19.431 m

Thinking:

This is a complex scenario requiring additional reasoning. (Terrain Description): The terrain ahead consists of a dirt path bordered by dense vegetation. The path appears to curve gently to the right. The surface looks like compacted dirt or gravel, which seems relatively firm. There are no visible large rocks, water, or other immediate obstacles on the path itself. However, the vegetation on either side is thick, suggesting it would be difficult to traverse beyond the path.

(Traversability Assessment):
- Dirt path (center-right, ~5 m ahead): traversable; the surface appears firm and suitable for driving. The curve suggests a gentle turn to the right.
- Vegetation border (left and right edges): non-traversable; the dense trees and underbrush would impede progress and could damage the vehicle if driven into.
- Potential hidden obstacles (path ahead): potentially traversable; while the path looks clear, there might be unseen dips, stones, or soft spots under the surface.

(Path Safety Reasoning): The safest path is along the dirt path as it curves to the right. The current velocity is relatively low (2.708 m/s), and the acceleration is moderate (0.517 m/s²), which allows for a controlled turn. The terrain appears stable, and there are no significant slopes or loose materials that could cause wheel slip. Given the curve in the path, steering right will align the vehicle with the natural direction of travel.

(Best Driving Action): sharp right with maintain speed

Given the need to follow the curve of the path and the absence of significant obstacles on the path itself, the best course of action is to sharply steer right while maintaining the current speed. This will ensure the vehicle stays on the safest part of the terrain and follows the intended path. The slight increase in speed can help maintain momentum through the curve without risking loss of control.

Action:
<offroad_action_776> <offroad_action_515> <offroad_action_776> <offroad_action_776> <offroad_action_776> <offroad_action_776>
<offroad_action_776> <offroad_action_776> <offroad_action_776> <offroad_action_776>

(b) Adaptive thinking

Figure 10: Planning and reasoning results of Offroad-VLA (PRL) on the TartanDrive dataset.

Across RELLIS-3D and TartanDrive, perception-centric baselines generally show limited alignment with terrain properties, with several methods exhibiting negative or near-zero RC. In contrast, OffRoad-VLA maintains positive RC on both datasets while achieving competitive terrain-aware action prediction, with PRL further improving TPA and terrain classification over SA. These results

indicate that OffRoad-VLA jointly captures action prediction and terrain-aware physical consistency across different off-road platforms. We next analyze the inference latency and reasoning-efficiency trade-offs between SA and PRL.

F.3 Qualitative Analysis on Unseen GOOSE Environments

To further assess out-of-distribution generalization, we qualitatively examine the behavior of OffRoad-VLA (PRL) on the GOOSE (Mortimer et al., 2024) dataset, which contains unseen environments, terrain types, and vehicle platforms. This evaluation complements the quantitative results by examining how well the terrain-aware policy transfers under a full distribution shift.

Figure 10 shows both DAG and AT in unseen scenes, demonstrating that Offroad-VLA (PRL) enables the model to adapt its reasoning and action generation to novel terrain conditions without fine-tuning. These observations highlight the ability of terrain-aware PRL training to generalize beyond the training distributions.

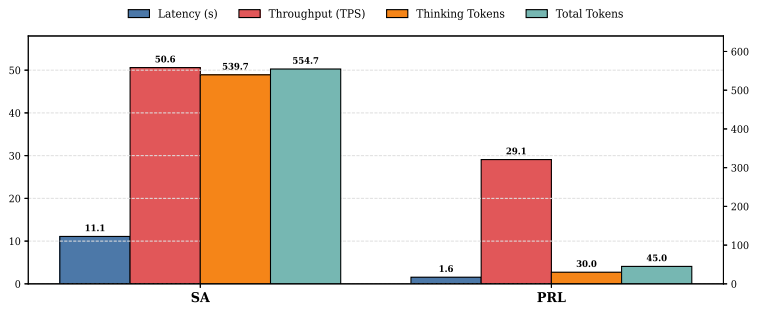


Figure 11: Inference latency and reasoning efficiency for SA and PRL variants, demonstrating that PRL achieves lower latency and more concise reasoning despite generating longer thinking chains.

F.4 Inference Latency and Reasoning Efficiency

A practical concern for onboard deployment is whether terrain-aware reasoning introduces prohibitive inference overhead. Fig. 11 compares latency, throughput, thinking tokens, and total tokens between the SA and PRL variants on TartanDrive, measured on a single GPU with greedy decoding. PRL reduces the average latency from 11.1 s to 1.6 s, a 7$\times$ speedup. This gain comes from far more concise reasoning. PRL generates only 30.0 thinking tokens compared to 539.7 for SA, a 94% reduction, and its total output shrinks from 554.7 to 45.0 tokens. PRL's throughput is lower (29.1 vs. 50.6 tokens/s) because the fixed cost of encoding the visual input and prompt is spread over far fewer generated tokens. Therefore, its efficiency gains stem from generating fewer tokens rather than faster decoding.

The CoT efficiency penalty in equation 15 successfully drives the model toward focused, action-oriented outputs. For both variants, the gap between thinking and total tokens consists almost entirely of the 10 action tokens, confirming that the models produce minimal extraneous output beyond the reasoning chain and the action sequence. Together, these results demonstrate that PRL not only improves planning quality and terrain safety but also substantially reduces computational overhead, bringing terrain-aware reasoning considerably closer to real-time onboard deployment.

G Closed-Loop Evaluation

Closed-loop evaluation tests whether terrain-aware planning can be translated into safe and executable vehicle control when predictions continuously affect the vehicle state and subsequent decisions (Shao et al., 2024a). Unlike open-loop evaluation, it captures the effects of accumulated prediction errors, state uncertainty, and terrain interactions during execution. We therefore evaluate OffRoad-VLA in four MAVS off-road environments (Hudson et al., 2020; Goodin et al., 2022; Carruth et al., 2024), each with two procedurally generated layouts, as shown in Fig. 12. The environments include **Hard Forest**, a gravelly dirt trail with scattered trees and grass; **Rough Desert**, a sandy trail with cracked

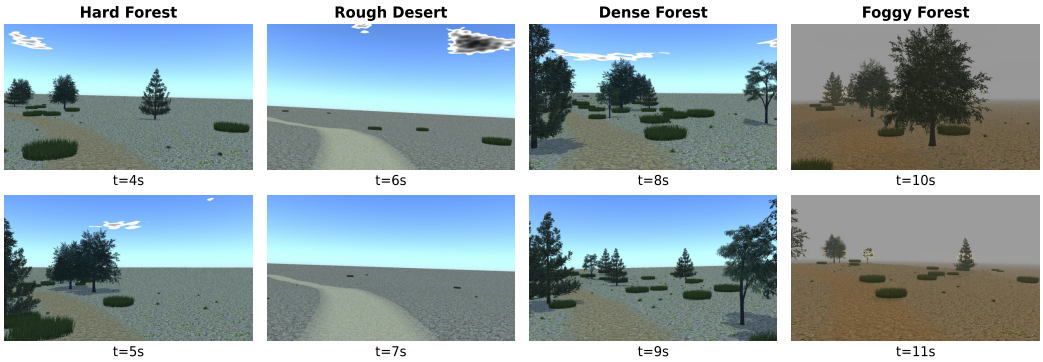


Figure 12: MAVS autonomous vehicle simulation environment.

and rocky terrain and sparse shrubs; **Dense Forest**, a vegetation-rich trail with frequent occlusions and strong shadows; and **Foggy Forest**, a forest trail with dense fog and limited visibility.

OffRoad-VLA replans at 2 Hz and predicts a 5-second trajectory with 10 waypoints at each step. The MAVS simulator uses a 0.01 s physics step and is paused during model inference, ensuring that inference latency does not affect the closed-loop dynamics. The predicted action tokens are decoded into continuous trajectories and tracked using a waypoint controller. Steering is proportional to the heading toward the waypoint 1.5 s ahead, while a PID controller tracks a target speed of 5 m/s. Each episode follows a 150 m trail route. An episode is considered successful when the vehicle reaches within 5 m of the goal after completing at least 80% of the route. An episode terminates upon rollover (roll or pitch $> 60^{\circ}$), collision, deviation of more than 20 m from the route, speed below 0.3 m/s for 5 consecutive seconds, or exceeding twice the nominal route time (60 s). No human intervention or trajectory correction is applied during evaluation.