**2026 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
AND TECHNOLOGY SYMPOSIUM
MODELING, SIMULATION, PROTOTYPING, AND VALIDATION (MSPV) TECHNICAL SESSION
AUG. 11-13, 2026 - NOVI, MICHIGAN**

MULTIMODAL DIGITAL-TWIN SIMULATIONS FOR OFF-ROAD AUTONOMOUS VEHICLES

Timothy Ross[1], **Julia Boone**[1], **and Fatemeh Afghah, PhD**[1]

[1]Clemson University, Clemson, SC

ABSTRACT

Research in autonomous driving has largely focused on structured, on-road environments in densely populated urban areas, leaving off-road autonomy comparatively underexplored despite its importance for applications such as search and rescue, agriculture, and defense. Progress in developing solutions for off-road autonomy is heavily constrained by the high cost, safety risks, and limited coverage of real-world off-road data collection, particularly for rare terrain-induced failure cases. To address these challenges, we introduce a large-scale simulation platform for off-road autonomous driving generated in the Unity3D engine. Our simulator provides indefinite data from multiple RGB cameras, LiDAR, GPS, and IMU sensors to support research in long-duration applications. Our platform also facilitates research in risk-aware planning through dynamic weather, lighting, and annotations for sequence-level failures like collisions and weather-induced sensor blockages. By combining long-duration temporal coverage, multimodal sensing, and procedurally generated terrain diversity, our simulator facilitates progress for systematic evaluation of learning-based autonomous driving models in unstructured, safety-critical environments.

Citation: T. Ross, J. Boone, F. Afghah, "Multimodal Digital-Twin Simulations for Off-Road Autonomous Vehicles," In *Proceedings of the 2026 Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*, NDIA, Novi, MI, Aug. 11-13, 2026.

1. INTRODUCTION

Autonomous driving research has progressed rapidly over the past decade, largely driven by the availability of large-scale multimodal datasets collected in structured urban environments. Benchmarks such as KITTI [1], nuScenes [2], Waymo Open [2], and BDD100K [3] have enabled major advances in perception, prediction, and planning for on-road autonomous vehicles. However, these datasets primarily focus on structured road networks with well-defined lanes, traffic rules, and relatively predictable surface conditions. In contrast, off-road autonomous driving remains underrepresented despite its importance in

DISTRIBUTION A. Approved for public release; distribution unlimited. OPSEC #: 10374 applications such as agriculture [4], forestry [5], disaster response [6], environmental monitoring [7], and defense operations [8]. This lack of off-road driving data exists in part due to the challenges associated with its collection [9]. Unlike urban environments, off-road terrains can present a large quantity of potential driving trajectories across numerous surfaces where traversability is not guaranteed. Additionally, environmental conditions vary rapidly and vehicle failure modes are often governed by the physical interaction with deformable or low-friction surfaces in such varied conditions. As a result, models trained on conventional on-road datasets often struggle to generalize to off-road conditions, particularly under distribution shifts in weather, lighting, and terrain composition. Furthermore, collecting real-world off-road data can be cost-prohibitive and operationally challenging, as it often involves specialized platforms, difficult terrain access, and elevated risk of vehicle damage or safety incidents.

From this, it is clear that a key barrier to progress in off-road autonomy is the lack of large-scale, multimodal, and temporally continuous data that captures the diversity and dynamics of natural terrain. While off-road datasets exist, they are typically limited in one or more aspects. For example, some datasets only include short temporal sequences (such as twenty second captures), which restrict the study of long-horizon planning, failure accumulation, and adaptation to gradually changing environmental conditions. Many contain static environmental conditions and do not encapsulate dynamic weather and lighting conditions, preventing evaluation under continuous and sometimes rapid distribution shift. Terrain diversity is also typically limited in current off-road datasets, only focusing on a single site or biome. Lastly, many do not contain synchronized multimodal sensing data (such as RGB, LiDAR, IMU, GPS, etc), which limits the ability to study sensor fusion and full-stack autonomy in physically grounded off-road scenarios.

To address these challenges, we present a large-scale, high-fidelity Unity-based simulation platform for unstructured, off-road autonomous vehicles. The platform is designed specifically to model the spatiotemporal complexity and environmental variability characteristic of off-road driving. This includes synchronized data from multiple RGB cameras, LiDAR, GPS, and IMU sensors for practically indefinite duration driving sequences in procedurally generated terrains. Unlike most existing benchmarks, our platform captures dynamic transitions in weather and lighting, enabling the study of perception and control robustness under continuous distribution shift. In addition to environmental diversity, our platform contains explicit designations for vehicle-terrain interaction failures, including collisions and terrain-induced epistemic failures (such as loss of traction or slope-induced immobility). These events are labeled at the sequence level, supporting research in risk-aware planning, failure prediction, and robustness evaluation, which are areas that are especially critical for safety in off-road deployments.

2. SCOPE OF AUTONOMOUS DRIVING DATASETS

On-road datasets. Autonomous vehicle perception and planning solutions have been primarily developed using large-scale, on-road datasets. Preliminary datasets like KITTI [1] established standardized evaluation techniques for stereo vision, optical flow, and 3D object detection in on-road driving scenarios. Recent datasets

Table 1: Summary of relevant driving datasets compared to our capabilities

Dataset	Off -Road	Real	Cont. Scenes	Long Scenes (1 min +)	Benign & Failures Sequences
CaT [10]	✓	✓		✓	
GO [11]	✓	✓		✓	
RELLIS-3D [12]	✓	✓	✓	✓	
KITTI-360 [13]					
SHIFT [14]			✓	✓	
AIODrive [15]					
SCOPE [16]					
CarlaScenes [17]					
KITTI [1]		✓			
nuScenes [2]		✓			
Waymo OPEN [18]		✓			
BDD100K [3]		✓			
OmniHD-Scenes [19]		✓			
Ithaca365 [20]		✓		✓	
PhysicalAI-Autonomous-Vehicles (NVIDIA) [21]		✓			
Ours	✓		✓	✓	✓

like nuScenes [2] and Waymo OPEN [18] expanded dataset scale and modality, providing initial synchronized camera, LiDAR, radar, and localization data across different cities and traffic conditions. While the risks associated with data collection in on-road environments are substantially lower than those in off-road driving, achieving sufficient coverage of real-world diverse locations, lighting, weather, and rare corner cases at scale remains costly and time-consuming, motivating the use of simulation and synthetic data generation. SCOPE [16] is an example of a synthetic, on-road, multimodal driving dataset collected from the CARLA [22] simulation environment. SCOPE features recordings from multiple collaborative vehicles and infrastructure sensors across diverse urban and highway scenarios. Similarly, CarlaScenes [17] is a synthetic dataset built in the CARLA simulator with a specific focus on challenging scenarios for visual and LiDAR-based odometry in autonomous driving. It provides synchronized camera, LiDAR, GNSS, and IMU data across diverse conditions such as slopes, dynamic traffic, and varying weather and lighting. Despite the breadth and scale of on-road driving datasets, they are fundamentally focused on structured, lane-based driving. Roads are well-defined, surface traversability is assumed, and scene dynamics are dominated by traffic agents. As a result, they provide limited exposure to the types of features that characterize off-road autonomy.

Off-road datasets. Datasets specifically targeting off-road environments are comparatively fewer and often more specialized compared to on-road datasets. For example, the Great Outdoors (GO) [11] dataset is a large-scale, real-world multimodal benchmark collected on a ground vehicle platform across various trails, forests, and open fields. While GO utilizes a comprehensive sensor suite (RGB, stereo, thermal, LiDAR, radar, IMU, and GPS), GO is collected in fixed real-world environments with largely static weather and lighting per run. RELLIS-3D [12] provides synchronized RGB and LiDAR data collected on a ground robot platform, but it is collected at a single site and does not model dynamic changes in weather or lighting conditions within its sequences. The CAVS Traversability (CaT) [10] dataset has a unique focus in terrain traversability for off-road driving, where terrain data is collected via monocular camera imagery in real off-road environments and trail regions are segmented according to the capability of different vehicle classes (sedan, truck, and off-road vehicles). However, CaT focuses on per-frame visual segmentation of static scenes and does not include multimodal sensing or long-horizon temporal sequences. Across these efforts, common limitations include restricted geographic diversity, short temporal coverage, and static environmental conditions, which make it difficult to evaluate robustness to continuous distribution shifts or long-term interaction with complex terrain. We summarize the key features (or lack thereof) of various on- and off-road driving datasets alongside the capabilities provided by our simulator in Table 1.

3. DATA GENERATION PIPELINE

Multimodal data is generated through a scalable simulation and data streaming pipeline designed to produce long-horizon, off-road driving sequences under dynamically changing environmental conditions. The pipeline consists of four main components: sensor simulation, procedural environment generation, scenario modeling, and real-time multimodal data collection. This process is summarized in Figure 1.

Table 2: Saved sequence directory organization

Modality	Path / Files
Camera	camera/{front,back,left,right,overhead}/ {scan_token.jpg}
GNSS	gnss/gnss_0/gnss.csv
IMU	imu/imu_0/imu.csv
LiDAR	lidar/lidar_0/{scan_token.npy}
Meta	sequence_config.json, uuid_timestamp.csv

3.1. Simulation Platform and Sensor Modeling

We build our data generation framework within the Unity3D (6.2) game engine using the open-source UnitySensors(2.0.5) [23] package. This package integrates 3D Lidar, various camera, IMU, and GNSS sensors into Unity3D, enabling physically grounded simulation of multiple sensing modalities commonly used in autonomous vehicles. Our simulated vehicle is equipped with five RGB cameras (front, back, left, right, overhead) for multi-view vision sensor data collection, a LiDAR sensor (modeled after the Velodyne VLP-32 sensor) for 3D point cloud data collection, a GNSS/GPS module for global positioning, and an IMU sensor for inertial and motion dynamics. Camera images are stored using lossy JPG compression (80% quality factor) to reduce storage overhead while preserving visual fidelity. LiDAR point clouds are stored as NumPy arrays with no header and compressed using fixed-precision encoding, reducing floating-point resolution while maintaining centimeter scale accuracy. Each LiDAR point consists four signed short values for the x, y, z, and intensity coordinates. GNSS and IMU data are recorded as time-stamped CSV files. Together, these sensors provide a synchronized multimodal representation of each driving sequence. Each sequence collected during data collection is stored using the file structure indicated in Table 2.

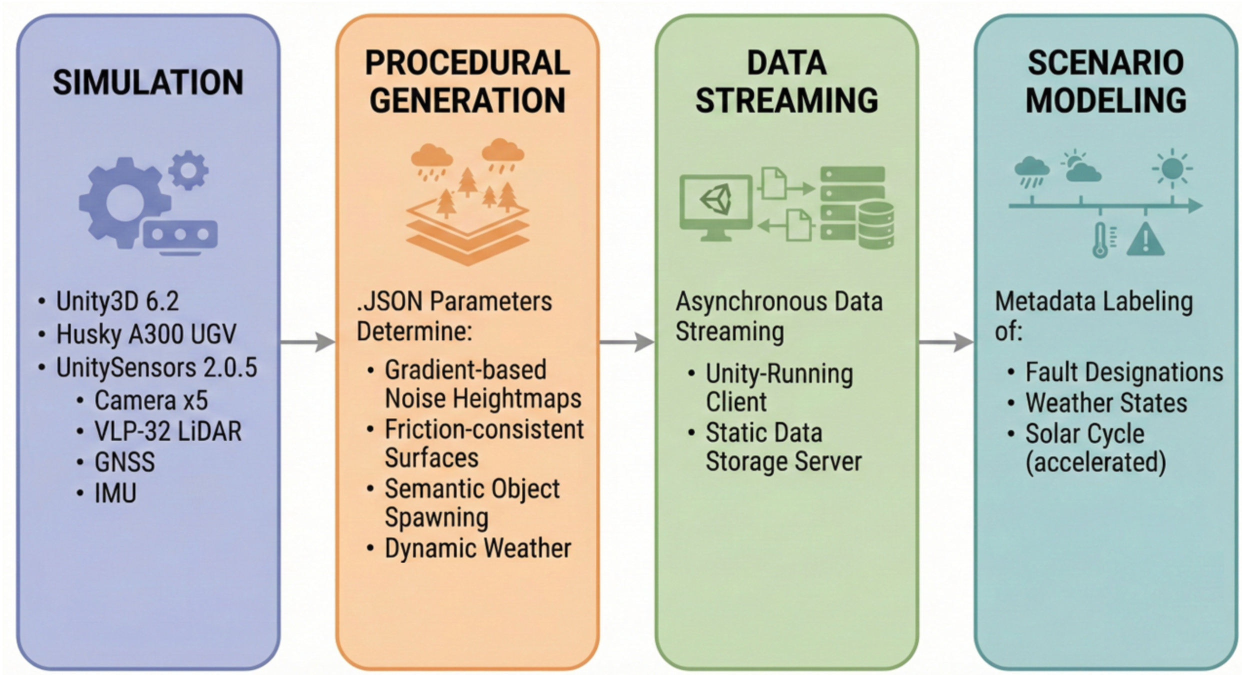

Figure 1: Overview of the data generation framework. Procedural configuration inputs, including random seed, biome selection, and weather parameters, guide Unity-based simulation of layered off-road environments with terrain heightmaps, surface materials, vegetation, and dynamic weather. A vehicle agent equipped with multimodal sensors (RGB cameras, LiDAR, GPS, IMU) collects synchronized data to statically store along with metadata and sequence-level failure annotations.

We integrate AutoDrive Simulator's [24] Clearpath Robotics Husky A300 robot model for navigation, which is a small rover-style unmanned ground vehicle (UGV). We utilize AutoDrive's default values for parameters that influence the Husky's dynamics (vehicle mass, wheel mass, wheel radius, suspension distance, etc). We also use AutoDrive Simulator's vehicle controller to teleoperate the vehicle.

3.2. Procedural Environment Generation

On-road and off-road datasets are both often limited in their intra-domain variance (e.g., across different parking lot configurations) and their inter-domain diversity (e.g., between highways and parking lots). We solve this issue by utilizing a seed-based procedural terrain generation pipeline that maps numerical and categorical input parameters into a physically realizable terrain. Each collection of parameters is designed to replicate a real-world biome. By deriving the terrain topology, including the distribution of vegetation and other objects, from a deterministic seed value, this generator can produce virtually infinite semantically identical yet spatially distinct environments.

It is necessary to briefly clarify the capabilities of Unity's native terrain system. Unity's terrain system provides a heightmap-based representation for creating large-scale landscapes, supporting elevation sculpting, texture painting, and the placement of vegetation such as trees and grass. While these tools facilitate the creation of highly-realistic environments, they are characterized by significant time investments by a skilled human. These characteristics make manual terrain authoring unideal for extensive autonomous vehicle evaluation where environment variance is critical. In addition, current generative approaches such as 3D-GPT [25] and SceneX [26] take significant time to create and do not generate at the scale required for long-duration evaluation. To solve this, we create an overhead framework to instruct Unity's terrain system to deterministically generate customized biomes through a manually-refined set of input parameters. When combined with an assortment of other Unity subsystems to create surface friction and environment conditions, we are able to generate environments that are realistic, large-scale, and require minimal time overhead. In our work, we specifically break the generation of terrains into four key components: heightmaps, surfaces, objects, and environmental conditions. We detail here what each component does and how we utilize them for our procedural generation pipeline.

Heightmaps. Heightmap diversity introduces large-scale variation in terrain elevation, slope, and surface curvature, enabling the simulation of hills, valleys, embankments, and uneven ground typical of natural off-road environments. In our work, Unity's heightmap-based terrain engine allows us to procedurally generate terrain elevations using a seeded Perlin or Simplex noise formulation. Perlin and Simplex noise are both gradient-dependent, which results in smooth, spatially coherent variations that resemble natural elevation patterns. Using Perlin noise formulation for terrain generation is a commonly used approach [27, 28, 29, 30], while Simplex noise shares significant conceptual overlap and is occasionally preferred for visual preferences. By manipulating the noise frequency or scale and adding additional noise fractal or domain warping, heightmap generation can be significantly diversified.

Surfaces. On-road terrains differ greatly from off-road in that, under normal environmental conditions, every paved route is considered traversable. In unstructured off-road environments, the traversability of a candidate path is contingent upon the topographic gradient and the friction between the vehicle's tires and the terrain surface. In our work, we replicate the effects of different surface structures within Unity3D via height value with blending factors to create realistic transitions. Each surface contains a unique tangential and perpendicular friction to ensure physical consistency between the same material (such as sand) in different environments. These friction values serve as a multiplier for the stiffness value of each wheel's WheelFrictionCurve, a proprietary Unity data structure that determines the friction properties of each Unity WheelCollider.

Objects. After terrain geometry and surface materials are generated, we procedurally populate the environment with vegetation and environmental detail objects. Using terrain-aligned texture or layer indices as semantic masks, our system spawns trees and small-scale details (grass, shrubs, ground clutter, etc) according to biome-specific configuration parameters. Placement routines are modular and driven by user-defined density, size, distribution, and object-type rules, ensuring that objects appear only in physically plausible regions (for example, vegetation on soil rather than rock or water). This material-aware spawning process increases environmental realism while preserving consistency between terrain appearance and obstacle distribution. Unity's colliders system, which are responsible for physics-based object and LiDAR raycast collisions, are rendered as capsules or boxes associated with their visual appearance instead of meshes to permit a larger quantity of objects to be rendered. Due to their datatype, detail objects (grass, shrubs) do not have colliders.

Environmental conditions. To model realistic distribution shifts within long-horizon sequences, we implement a deterministic weather and illumination controller in Unity. At the start of each sequence, the global directional light is randomized (via a seed) to vary opposing sun and moon position. The sun and moon angle is then incrementally rotated over time to simulate gradual lighting change. The transition from day-to-night is accompanied by the interpolation of temporal-specific lighting and weather conditions (e.g., cloud color) to ensure real-world authenticity. Weather is modeled as a discrete-state process evaluated at fixed intervals, where transitions occur probabilistically and are smoothed using time-based interpolation. Rain is represented through a particle system, fog through Unity's global fog density, and wind through a configurable wind zone; cloud opacity is adjusted via material alpha to visually reflect weather intensity. This design enables continuous, physically plausible transitions in environmental conditions within a single driving sequence.

3.3. Real-Time Multimodal Data Streaming

Generating long-horizon multimodal data introduces significant memory and I/O challenges as storing all synthetically created sensor data in RAM for the duration of a scene degrades physics engine and data generation performance. Alternatively, writing synthesized data to disk during a scene causes frequent, lengthy delays. To address this, we implement a client–server streaming architecture between two machines. The data streaming pipeline uses a unique half-duplex, asynchronous communication protocol between a Unity-running client and a static data storage server instantiated in Python using the socket library. Each frame recorded from Unity3D by the client is encapsulated into a packet with a unique 14-byte header and a payload of the sensor data. Within each header, the frame is assigned a randomly generated Universally Unique Identifier (UUID), a scan token that is utilized on the server's side to help organize the intra-frame sensor readings, an associated sensor type byte that ensures the data is properly formatted based on the type of sensor being sent, and a sensor ID that is used when there are multiples of a particular sensor type (primarily RGB cameras) that need to be distinguished between. This packet format is shown in Figure 2.

Communication begins with an initialization packet containing the sequence UUID, streaming parameters (scan frequency and frame count), and scenario metadata such as weather and lighting conditions. Sensor data packets are then buffered in memory for short intervals before being transmitted to the server, preventing simulation slowdowns during computationally intensive scenes (caused by things such as heavy weather or dense object rendering). Once the total number of frames has been processed on the server side, the user is prompted for a fault designation. The fault designation marks the end of that sequence. After receiving a final acknowledgment packet from the server indicating a fault designation was chosen, the car object is relocated, the weather is randomized, and a new sequence is started with a new initialization packet. This allows for seamless and continuous simulation in the current environment with varied environmental conditions.

3.4. Scenario Modeling

Unlike many existing datasets that capture short or static driving segments, our simulator emphasizes long-duration sequences with

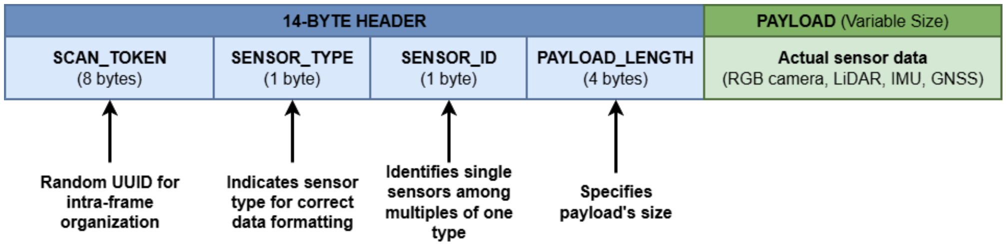

Figure 2: Packet format for real-time sensor streaming. A 14-byte header encodes a per-frame scan token (UUID), sensor type/ID, and payload length, followed by a variable-size payload containing synchronized multimodal sensor data.

evolving environmental conditions. During a single driving episode, weather intensity, lighting direction, and terrain interaction may change continuously. While each sequence has an associated meta-data file, sequences are also manually annotated with event-level fault designations. The event-level designations include:

- **Benign**: in which the vehicle had normal traversal and experienced no collisions
- **Collision**: in which the vehicle had abnormal traversal and experienced collisions with objects in the scene
- **Epistemic Failure**: in which the vehicle experienced terrain-induced immobilization (such as entrapment upon encountering a slope or low-friction surface) or navigational divergence, where the vehicle selects a candidate path that exceeds physical or tractive capabilities,
- **Water**: in which the vehicle was immobilized by driving through a water plane
- **Weather-Associated Failure**: in which a collision occurred during adverse weather or poor lighting conditions

Our simulator contains options for both static and dynamic weather conditions. Dynamic weather contains randomized transition times and speeds alongside a gradual interpolation of environmental factors that differ between the two states. The simulator contains five distinct preset environmental states:

- **Clear**: characterized by minimal cloud cover and negligible wind
- **Foggy**: in which visibility is significantly limited
- **HeavyRain**: defined by dense precipitation that obstructs camera views and a substantial increase in wind
- **LightRain**: featuring moderate rain that lightly obstructs visibility and a slight increase in wind
- **Scattered**: consisting of moderate cloud distribution

Environmental conditions are stochastically initialized for each sequence, ensuring generated conditions are semantically and geologically appropriate for a chosen environment. Users can add additional weather conditions through a UI by adjusting preset variables like fog density and precipitation rate. The interpolation to and from any user-created conditions is handled automatically by the simulation's weather controller.

Finally, our simulator is capable of generating sequences representing every time of the day. Metadata labels are provided based upon solar elevation angle (θ), where

$0°$ denotes the horizon at sunrise. The classifications are defined as follows: **Day** ($10° < \theta \leq 170°$), **Dusk** ($170° < \theta \leq 190°$), **Night** ($190° < \theta \leq 350°$), and **Dawn** ($350° < \theta \leq 10°$). Sequences containing multiple different classifications are listed within the meta-data in order of appearance (e.g., "Night, Dawn"). Initial solar position is randomized and contains a modifiable acceleration factor to differently model changing lighting conditions within the sequence's limited temporal window.

The lighting and weather variability introduced here can constitute adverse weather conditions if they induce significant perceptual degradation. The Foggy, HeavyRain, and LightRain weather conditions, as well as the Night lighting condition, are determined significant enough to constitute a weather-associated failure fault designation. Figure 3 demonstrates how these conditions affect a multi-camera perception system.

4. CURRENT RESULTS

We choose to create a set of 7 different environments to demonstrate the diversity provided by our environment generation system. These environments are synthesized by human-refined input parameters described in Section 3.2 and are designed to introduce challenges unique to unstructured, off-road navigation. A visual sample of these environments is shown within Figure 4 and are defined as follows:

- **DenseMuddyForest**: A high-density, forest biome characterized by gradual but significant elevation gradients and low-elevation mud deposits.
- **Desert**: An arid landscape defined by high-relief sand dunes. Challenges vehicle traversal due to extreme tractive limitations on non-cohesive sand and path-planning within a topographically featureless environment.
- **Lakes**: A temperate woodland biome dominated by large bodies of water. Water-hazard avoidance remains critical as a potential unrecoverable failure.
- **NormalForest**: A temperate woodlands characterized by moderate tree density and sparse vegetation. Serves as a baseline for object detection and traversal tasks.
- **Savanna**: A grassland biome comprising of tall, dense vegetation that obscures embedded rocky obstacles. Limited onboard visibility tests object recognition of overhead camera
- **Taiga**: A subarctic forest characterized by high-relief topography and extreme tractive limitations. At low elevations, mud deposits create significant risk of entrapment.
- **Wetlands**: A humid domain featuring tall, dense grass and persistent shallow water beds. Limited onboard visibility tests scene segmentation of overhead camera. Like the Lakes biome, water failure is a potential unrecoverable failure.

To demonstrate the efficacy of our sensor simulation platform, we collect a variety of short (20s), medium (180s) and long-duration (1,500s) sequences sampled at 2 Hz among the seven distinct environments. Varying onboard and flyover camera perspectives from these environments are shown within Figure 5.

5. CONCLUSION

We introduced dynamic simulator for off-road autonomous driving capable of long

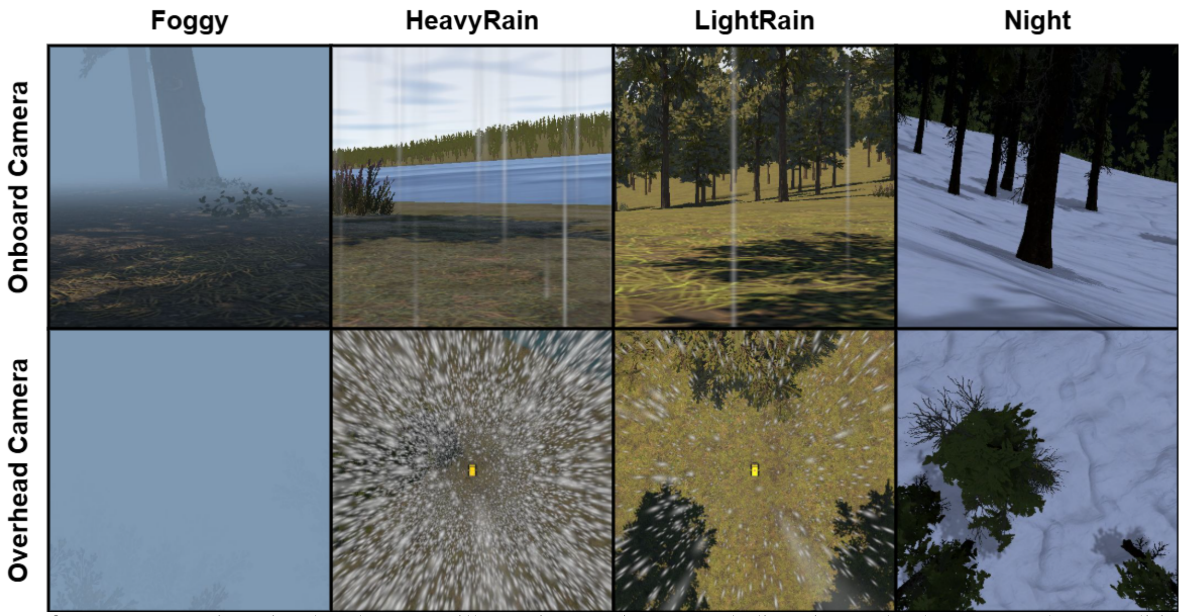

Figure 3: Representative simulated scenes illustrating environmental diversity and adverse weather conditions. Examples introduce substantial visual variability and degradation and include fog, rain, and an absence of lighting across forested, and lakeside settings, shown from both ego-view and overhead perspectives.

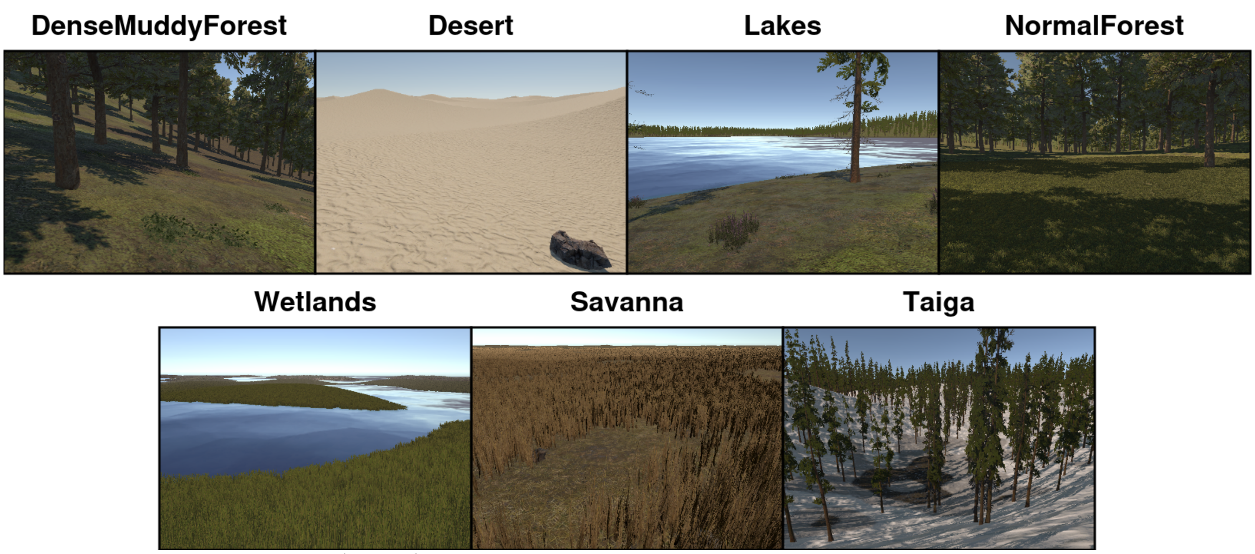

Figure 4: Visuals of all seven user-defined environments

duration, high-resolution sequences. Our integrated data generation pipeline, including high-fidelity sensor modeling, procedural environment generation, multimodal data streaming, and structured scenario modeling facilitates the collection of long-horizon sequences across a variety of uniquely generated yet spatially distinct environments. By incorporating dynamic weather and lighting conditions in collaboration with sequence-level fault designations, our simulation platform provides the robustness necessary to generalize to a broad spectrum of off-road, unstructured autonomous applications.

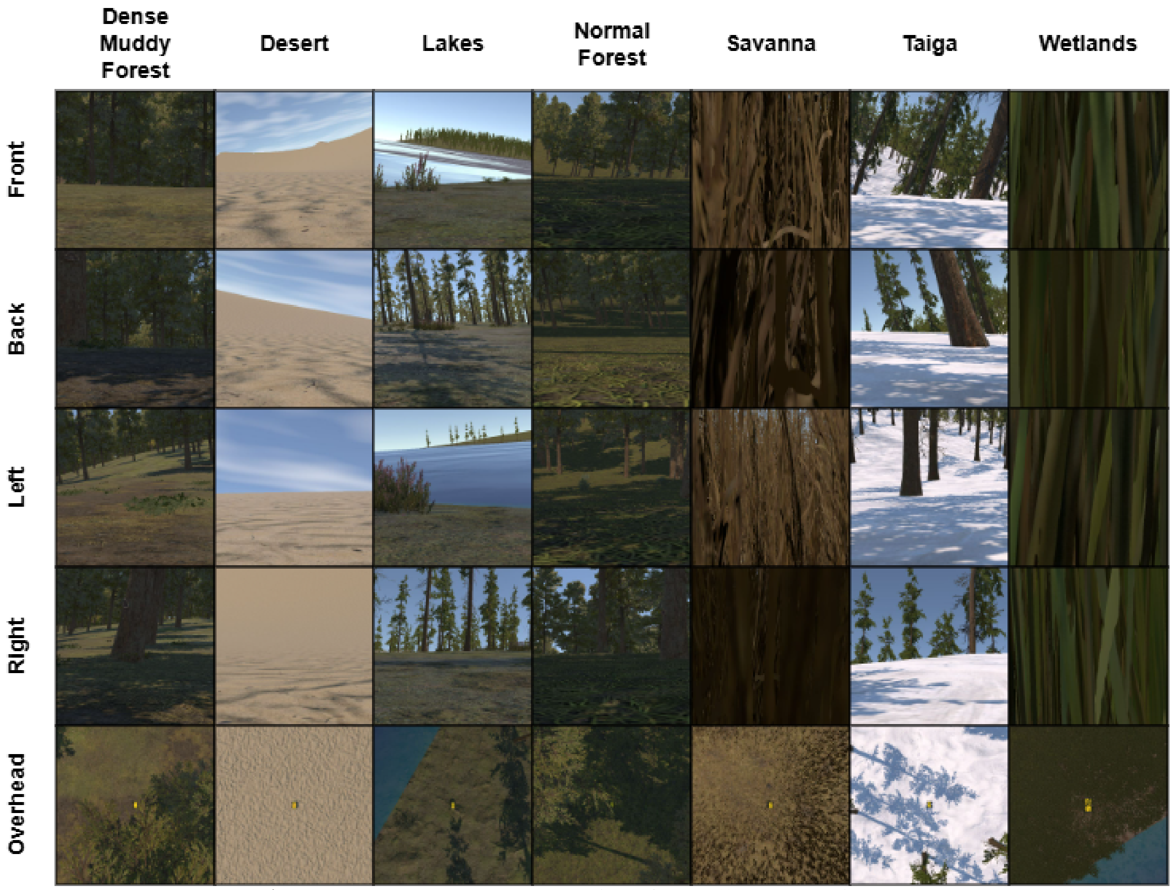

Figure 5: RGB camera perspectives from all seven environments

6. FUTURE WORK

Future work will explore updates to the LiDAR sensor profile to minimize real-to-sim differences. As stated in Section 3.2, the utilization of physics colliders to determine LiDAR raycast points introduces unique challenges for otherwise collision-free objects like grass. Alternatively, mesh renderer-based raycasting may better model the high geometrical complexity of objects like grasses, trees, and rocks while preventing the integration of computationally expensive mesh colliders. Furthermore, the complex nature of terramechanics, particularly concerning non-cohesive surfaces like snow and sand, presents a significant modeling challenge. Surface deformation of these substrates under vehicle load may be difficult to reliably replicate in the real-world. Thus, a high-fidelity synthetic simulation of these interactions can further advance off-road autonomous driving.

References

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The international journal of robotics research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 618–11 628.

[3] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, “Bdd100k: A diverse driving dataset for heterogeneous multitask learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636–2645.

[4] Y. Bai, B. Zhang, N. Xu, J. Zhou, J. Shi, and Z. Diao, “Vision-based navigation and guidance for agricultural autonomous vehicles and robots: A review,” *Computers and Electronics in Agriculture*, vol. 205, p. 107584, 2023.

[5] F. A. Ruetz, N. Lawrance, E. Hernández, P. V. K. Borges, and T. Peynot, “Foresttrav: 3d lidar-only forest traversability estimation for autonomous ground vehicles,” *IEEE Access*, vol. 12, pp. 37 192–37 206, 2024.

[6] R. Allen and M. Mazumder, “Toward an autonomous aerial survey and planning system for humanitarian aid and disaster response,” in *2020 IEEE Aerospace Conference*, 2020, pp. 1–11.

[7] F. Afghah, A. Razi, J. Chakareski, and J. Ashdown, “Wildfire monitoring in remote areas using autonomous unmanned aerial vehicles,” in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 835–840.

[8] D. Patil, M. Ansari, D. Tendulkar, R. Bhatlekar, V. N. Pawar, and S. Aswale, “A survey on autonomous military service robot,” in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, 2020, pp. 1–7.

[9] D. W. Carruth, C. T. Walden, C. Goodin, and S. C. Fuller, “Challenges in low infrastructure and off-road automated driving,” in *2022 Fifth International Conference on Connected and Autonomous Driving (MetroCAD)*, 2022, pp. 13–20.

[10] S. Sharma, L. Dabbiru, T. Hannis, G. Mason, D. W. Carruth, M. Doude, C. Goodin, C. Hudson, S. Ozier, J. E. Ball, and B. Tang, “Cat: Cavs traversability dataset for off-road autonomous driving,” *IEEE Access*, vol. 10, pp. 24 759–24 768, 2022.

[11] P. Jiang, K. Viswanath, A. Nagariya, G. Chustz, M. Wigness, P. Osteen, T. Overbye, C. Ellis, L. Quang, and S. Saripalli, “Go: The great outdoors multimodal dataset,” 2025. [Online]. Available: https://arxiv.org/abs/2501.19274

[12] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, “Rellis-3d dataset: Data, benchmarks and analysis,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1110–1116.

[13] Y. Liao, J. Xie, and A. Geiger, “Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3292–3310, 2023.

[14] T. Sun, M. Segu, J. Postels, Y. Wang, L. Van Gool, B. Schiele, F. Tombari, and F. Yu, “Shift: a synthetic driving dataset for continuous multi-task domain adaptation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 21 371–21 382.

[15] X. Weng, Y. Man, D. Cheng, J. Park, M. O'Toole, and K. Kitani, "All-In-One Drive: A Large-Scale Comprehensive Perception Dataset with High-Density Long-Range Point Clouds," *arXiv*, 2020.

[16] J. Gamerdinger, S. Teufel, P. Schulz, S. Amann, J.-P. Kirchner, and O. Bringmann, "Scope: A synthetic multi-modal dataset for collective perception including physical-correct weather conditions," in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, 2024, pp. 1–8.

[17] A. Kloukiniotis, A. Papandreou, C. Anagnostopoulos, A. Lalos, P. Kapsalas, D.-V. Nguyen, and K. Moustakas, "Carlascenes: A synthetic dataset for odometry in autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4520–4528.

[18] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[19] L. Zheng, L. Yang, Q. Lin, W. Ai, M. Liu, S. Lu, J. Liu, H. Ren, J. Mo, X. Bai *et al.*, "Omnihd-scenes: A next-generation multimodal dataset for autonomous driving," *arXiv preprint arXiv:2412.10734*, 2024.

[20] C. A. Diaz-Ruiz, Y. Xia, Y. You, J. Nino, J. Chen, J. Monica, X. Chen, K. Luo, Y. Wang, M. Emond *et al.*, "Ithaca365: Dataset and driving perception under repeated and challenging weather conditions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 21 383–21 392.

[21] N. Corporation, "Physicalai-autonomous-vehicles," https://huggingface.co/datasets/nvidia/PhysicalAI-Autonomous-Vehicles, 2025.

[22] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 13–15 Nov 2017, pp. 1–16. [Online]. Available: https://proceedings.mlr.press/v78/dosovitskiy17a.html

[23] R. Tanaka, A. Harada, and the Field Robotics Japan contributors, "Unitysensors," https://github.com/Field-Robotics-Japan/UnitySensors, 2025.

[24] T. V. Samak, C. V. Samak, and M. Xie, "Autodrive simulator: A simulator for scaled autonomous vehicle research and education," in *Proceedings of the 2021 2nd International Conference on Control, Robotics and Intelligent System*, ser. CCRIS '21. New York, NY, USA: Association for

Computing Machinery, 2021, p. 1–5. [Online]. Available: https://doi.org/10.1145/3483845.3483846

[25] C. Sun, J. Han, W. Deng, X. Wang, Z. Qin, and S. Gould, "3d-gpt: Procedural 3d modeling with large language models," 2024. [Online]. Available: https://arxiv.org/abs/2310.12945

[26] M. Zhou, Y. Wang, J. Hou, S. Zhang, Y. Li, C. Luo, J. Peng, and Z. Zhang, "Scenex: Procedural controllable large-scale scene generation," 2024. [Online]. Available: https://arxiv.org/abs/2403.15698

[27] A. Jain, A. Sharma, and Rajan, "Adaptive & multi-resolution procedural infinite terrain generation with diffusion models and perlin noise," in *Proceedings of the Thirteenth Indian Conference on Computer Vision, Graphics and Image Processing*, ser. ICVGIP '22. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3571600.3571657

[28] S. Zhang, Z. Xu, Z. Liu, and M. Liu, "Noise algorithms in game terrain generation," in *2025 10th International Conference on Image, Vision and Computing (ICIVC)*, 2025, pp. 611–618.

[29] K. S. Emmanuel, C. Mathuram, A. R. Priyadarshi, R. A. George, and J. Anitha, "A beginners guide to procedural terrain modelling techniques," in *2019 2nd International Conference on Signal Processing and Communication (ICSPC)*, 2019, pp. 212–217.

[30] T. Archer, "Procedurally generating terrain," in *44th annual midwest instruction and computing symposium, Duluth*, 2011, pp. 378–393.

7. ACKNOWLEDGMENT

This work was supported by Clemson University's Virtual Prototyping of Autonomy Enabled Ground Systems (VIPR-GS), under Cooperative Agreement W56HZV-21-2-0001 with the US Army DEVCOM Ground Vehicle Systems Center (GVSC). This research used in part resources on the Palmetto Cluster at Clemson University under National Science Foundation awards MRI 1228312, II NEW 1405767, MRI 1725573, and MRI 2018069. The views expressed in this article do not necessarily represent the views of NSF or the United States government.