Further experiments are being conducted for the paper to be submitted to IROS25
This paper is not available in Pre-Print currently

HyperVL-Sim: Accelerating Autonomy with Hyper-Realistic Vehicle-in-Loop simulation for Sim2Real Transfer Learning

Aayush Agrawal
Department of Chemical Engineering,
Indian Institute of Technology Madras
Chennai, India.
agrawal@smail.iitm.ac.in

Qiranul Saadiyean
Department of Aerospace Engineering,
Indian Institute of Science, Bangalore
Bangalore, India.
qiranuls@iisc.ac.in

Suresh Sundaram
Department of Aerospace Engineering,
Indian Institute of Science, Bangalore
Bangalore, India.
vssuresh@iisc.ac.in

***Abstract*—While simulation holds immense promise for developing robust algorithms for autonomous vehicles in safety-critical situations, current limitations in photorealism and the absence of comprehensive Vehicle-in-the-Loop (VIL) testing with diverse sensors hinder its effectiveness. This paper introduces "HyperVL-Sim," a hyper-realistic VIL simulator designed for autonomous vehicles. HyperVL-Sim integrates various sensor modalities and facilitates Sim2Real transfer learning for training computer vision models. We showcase the power of HyperVL-Sim by demonstrating its deployment on a Level 5 autonomous vehicle for high-level software testing. With HyperVL-Sim, we integrate actual autonomous vehicle hardware, mirroring real-world constraints and minimizing the Sim2Real domain gap experienced during deployment. This includes realistic constraints regarding available processing power and actual hardware latency, crucial for developing software that functions effectively on the target platform. HyperVL-Sim also enables the creation of critical scenarios that are too rare or dangerous to be replicated in the real world, facilitating rigorous testing for a truly safe self-driving future. Further we demonstrate HyperVL-Sim for downstream task and achieved an mAP of 0.583 on a pure Sim2Real object detection task. With Sim2Real transfer learning, we improved this to an mAP of 0.904, which is only 4% less than the model trained on purely real data, demonstrating the effectiveness of reducing the real dataset size by 50% while maintaining high performance.**

Fig. 1. HyperVL-Sim map imported inside the simulation environment.

I. Introduction

Recent advancements in simulation have emerged as an essential tool for advancing new algorithms in robot perception, learning, and evaluation [1], [2], [3]. For safety-critical domains, such as autonomous vehicles, testing in simulation is always preferred than direct operation in the physical world. Simulation offers the potential to rapidly synthesize novel data for training, including challenging edge cases difficult to capture in the real world [4], [5]. Thus, simulation could enable the development of algorithms and models better equipped to handle the diverse challenges of the physical world, facilitating their deployment on embodied mobile agents.

Another advantage of simulation is Sim2Real transfer learning, where training autonomous (e.g., virtual robots) in diverse, realistic simulators, followed by transferring the learned skills to real world [6]. Despite the potential of simulation, the stark lack of photorealism and the paucity of diverse high fidelity simulated sensor representations, remain crucial barriers to realizing this promise for autonomous driving tasks. Thus, there is a need for high-fidelity simulation systems that include all the necessary sensors and minimize the Sim2Real domain gap for autonomous vehicle training and testing.

Simulators are primarily used for Software-in-the-Loop (SIL) testing of algorithms because the vehicle, sensor, actuator, and controller all consist of virtual models. However, in simulation-based testing, there may be errors between the behavior of the actual vehicle and its dynamic characteristics due to the application of virtual vehicle dynamics models and the occupants being unable to feel the vehicle's behavior. To ensure driver safety, an effective safety verification method must be developed by testing the systems over millions or even billions of kilometers [7].

To address these challenges, researchers have developed Vehicle-in-the-Loop (ViL) simulation [8], [9], [10], [11]. ViL involves integrating the real vehicle system into a virtual clone simulated in real-time. In this setup, a virtual environment is simulated, along with virtual sensors attached to the clone of the real vehicle. The detected simulator sensor data is then transmitted back to the real vehicle as the control input to the actuators. This allows the vehicle to react to the injected

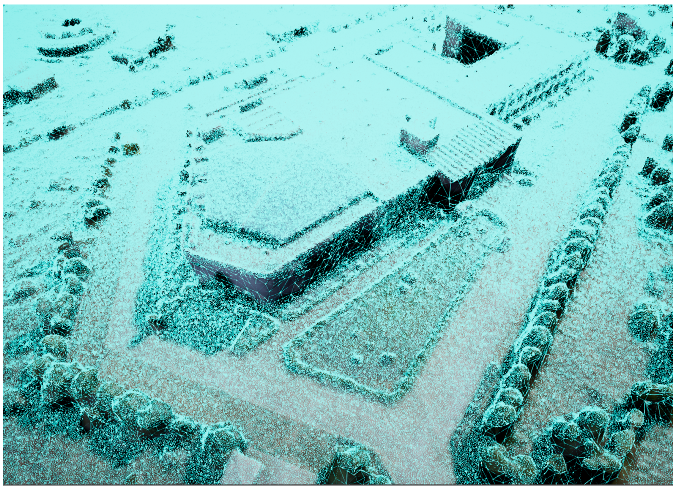

Fig. 2. Mesh density visualization from HyperVL-Sim. The mesh from fused point cloud has high mesh density leading to a smoother mesh after triangulation. This leeds to smoother simulation as compared with inaccurate mesh

sensor data in closed-loop operation. However, none of the existing open-source simulation systems offer VIL testing. The existing closed-source ViL simulations are of low fidelity and lack most of the essential autonomous vehicle sensors for data generation and transfer learning.

To address these limitations, we introduce HyperVl-Sim: A novel hyper-realistic simulation platform, built on top of Unreal Engine 4, designed to bridge the gap between simulation and reality. HyperVL-Sim incorporates a comprehensive suite of sensors, enabling accurate perception modeling and facilitating sim-to-real transfer learning. By integrating with actual autonomous vehicle hardware, HyperVL-Sim supports Vehicle-in-the-Loop (VIL) simulation, allowing for rigorous testing and validation under real-world conditions. This enables the verification of control systems and chassis control systems through vehicle testing, reflecting the dynamic characteristics of actual vehicles along with the algorithm validation.

A key innovation in HyperVL-Sim is the ability to generate highly realistic environments efficiently through multiple camera views. This eliminates the need for manual creation of individual components, accelerating environment development and enhancing simulation fidelity.

We demonstrate the effectiveness of HyperVL-Sim through its application to lane detection and object detection tasks, showcasing its potential to accelerate the development of robust autonomous driving algorithms. In summary, the contributions of this paper are as follows:

- HyperVl-Sim: A novel hyper-realistic vehicle in loop simulation platform for autonomous vehicle testing.
- High-Fidelity Map Generation: Utilized Structure from Motion (SfM) techniques to generate high-fidelity maps by fusing aerial and ground point clouds, providing a detailed and accurate simulation environment.
- Real-World Validation: Demonstration of HyperVL-Sim's effectiveness through successful deployment on a Level 5 autonomous vehicle and testing algorithms with vehicle in the loop.
- Bridging the Sim-to-Real Gap: Demonstration of effective Sim2Real transfer learning using HyperVL-Sim, leading to improved performance in object detection and semantic segmentation tasks.
- Enhanced Model Generalization: Demonstrate the ability of simulator to generate high-fidelity data, resulting in models that generalize better to real-world conditions compared to traditional simulation environments and also reduce the need for real data.

II. Related Work

Hardware in the Loop (HIL) testing is a concept that has been widely explored in multiple industries. The method originated in the 1950s and 1960s, and it found primary use in the aerospace industry and developing flight control systems [12], [13]. Later, due to advancements in electronic technology, HIL systems became more prevalent due to affordability. These days, HIL is widely used in many areas aerospace, automotive, electronics, energy, etc.

In the automotive industry, Hardware-in-the-Loop (HIL) testing is a crucial process for validating the functionality and performance of electronic control units (ECUs) and other vehicle components. [14] presented a method for ensuring stable and precise voltage regulation in standalone voltage-sourced converters. The proposed robust backstepping controller is designed to handle uncertainties and disturbances, and its effectiveness is validated through Hardware-in-the-Loop (HIL) testing. In [15] presented a real-time HIL simulation platform for electric vehicle powertrains. Soeiro et. al in [16] addressed the development of a HIL system to take into account the power systems of vehicles. HIL testing is a critical component of the autonomous vehicle development process [8], [17], [18]. By simulating the complex sensors, actuators, and environmental conditions that autonomous vehicles must navigate, HIL allows developers to thoroughly test and validate the vehicle's control systems in a safe, repeatable, and cost-effective manner.

In [8] Brogle et. al explored the simulation of high-level vehicle sensors such as cameras and lidar in Carla, emphasizing the absence of real-time constraints. This flexibility allowed simulations to vary time steps, enabling detailed scenario analysis and algorithm testing. The integration of CARLA with ROS is a significant contribution, verifying sensor measurement correlation between real and simulated environments without real-time limitations. This approach demonstrated efficiencies gained through simulation experiments, reducing the necessity for initial real vehicle tests.

Fathy et. al in [17] highlighted the evolution of Hardware-in-the-Loop (HIL) simulation from a tool for control prototyping to a comprehensive system modeling and synthesis paradigm. It introduced an advanced Engine-in-the-Loop (EIL) simulation facility that employs HIL techniques for system-level experimental evaluation of powertrain interactions. It enables the validation of performance and fuel efficiency predictions

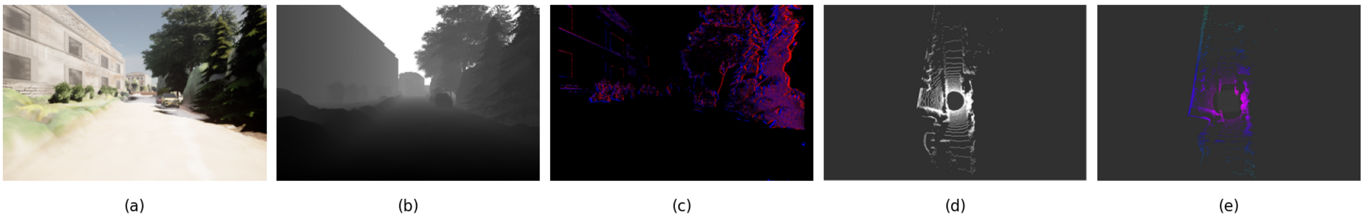

Fig. 3. Various sensor modalities from HyperVL-Sim(a) Optical Camera (b) Depth Camera (c) Event Camera (d) LiDAR Point Cloud (e) Semantic LiDAR Point Cloud

across various conventional and hybrid powertrains. In [19], a crowd-sourced framework is proposed that utilizes extensive image data from production vehicles to reconstruct large-scale 3D scenes using NeRF, addressing data quality challenges through filtering and a structured 3D reconstruction pipeline. [20] provides a comprehensive survey of NeRF applications in Autonomous Driving (AD), categorizing them into perception, 3D reconstruction, SLAM, and simulation, and summarizing key findings. [21] reviews methods for bridging the reality gap in autonomous driving, focusing on sim2real, digital twins, and parallel intelligence technologies, and discusses future directions, including the integration of large models and improved algorithms. [22] shows that simplifying action spaces and reward functions enhances sim2real performance in AWS DeepRacer. [23] introduces the AADS system, which augments real-world images with simulated traffic to create photorealistic, fully annotated simulation data, improving scalability and realism in AD simulations. Lastly, [4] presents the RarePlanes dataset, a large open-source collection combining real and synthetic satellite imagery to improve aircraft detection and classification, demonstrating the value of synthetic data in enhancing overhead imagery analysis. Most of the existing works focus on simulation for autonomous vehicles and Vehicle-in-the-Loop (VIL) testing for hardware validation. [24] Developed a way of high fidelity simulation, but it cannot be modified to create custom scenarios for our need. Hence there is a lack of high-fidelity Software-in-the-Loop (SIL) simulation frameworks that integrate hardware constraints. Our HyperVL-Sim addresses this gap by:

- Providing a high-fidelity simulation platform for Sim2Real transfer learning, including VIL simulation that accurately mirrors real-world hardware constraints.
- Supports downstream tasks such as object detection and segmentation.
- HyperVL-Sim offers the potential to generate real-life SLAM maps within the simulation and execute path planning directly, further enhancing its applicability in autonomous driving development and research.

III. Architecture

A. Map Generation

Given a log with camera images, our goal is to construct a digital twin, from which we can generate our map. In this study, we utilized two distinct sets of image data: one captured from a ground vehicle and the other from a drone. This dual-source approach was employed to enhance the accuracy of road feature detection, as roads may not always be clearly visible from the drone's aerial perspective due to obstacles, shadows, or limited resolution.

Structure from motion (SfM) based method is adopted to generate hyper-scale three-dimensional (3D) models using overlapping images acquired from different perspectives with standard optical cameras [25] [26]. SfM requires point correspondences between images, which is extracted using common feature extraction pipelines like SHIFT as seen in Fig.5. Photogrammerty based SfM method is used to generate our Hyper Sim map. The mesh from HyperVL-Sim is compared with the current state of the art 3d reconstruction method called Gaussian Splatting (3DGS), using FID (Fréchet Inception Distance) and LPIPS (Learned Perceptual Image Patch Similarity) metrics [27]. While FID measures the distance between feature distributions of generated and real images, assessing how similar the two distributions are, LPIPS compares perceptual similarity between images by analyzing features extracted from deep neural networks [28].

As seen from Table I, photogrammery based 3D mesh has better visual features along with smoother mesh. This helps in minimizing the domain gap during Sim2Real transfer learning.

We employed the SIFT (Scale-Invariant Feature Transform) algorithm to detect and describe keypoints within each image. Images were captured by a drone for aerial perspectives, denoted as I_d, and by a ground vehicle for detailed road views, denoted as I_g, ensuring complementary coverage. We applied the SIFT algorithm to extract keypoints from both image sets. For each image I_i, keypoints were identified as $u_{i1}, u_{i2}, \ldots, u_{in}$. Correspondences between drone and ground images were established, forming pairs (u_{ij}, u_{kl}). Using the matched keypoints, we computed the fundamental matrix F for each image pair: $u_{ij}^T F u_{kl} = 0$ The essential matrix E is derived as: $E = K^T F K$ where K is the intrinsic calibration matrix. The 3D points were reconstructed by minimizing the reprojection error:

$$X_i = \text{argmin}_X \sum_j \|u_{ij} - P_j X\|^2$$

resulting in two point clouds: P_d from the drone and P_g from the ground vehicle as seen in Fig. 5.

The 3D points X_i were reconstructed by minimizing the reprojection error: By following these steps, SfM algorithms

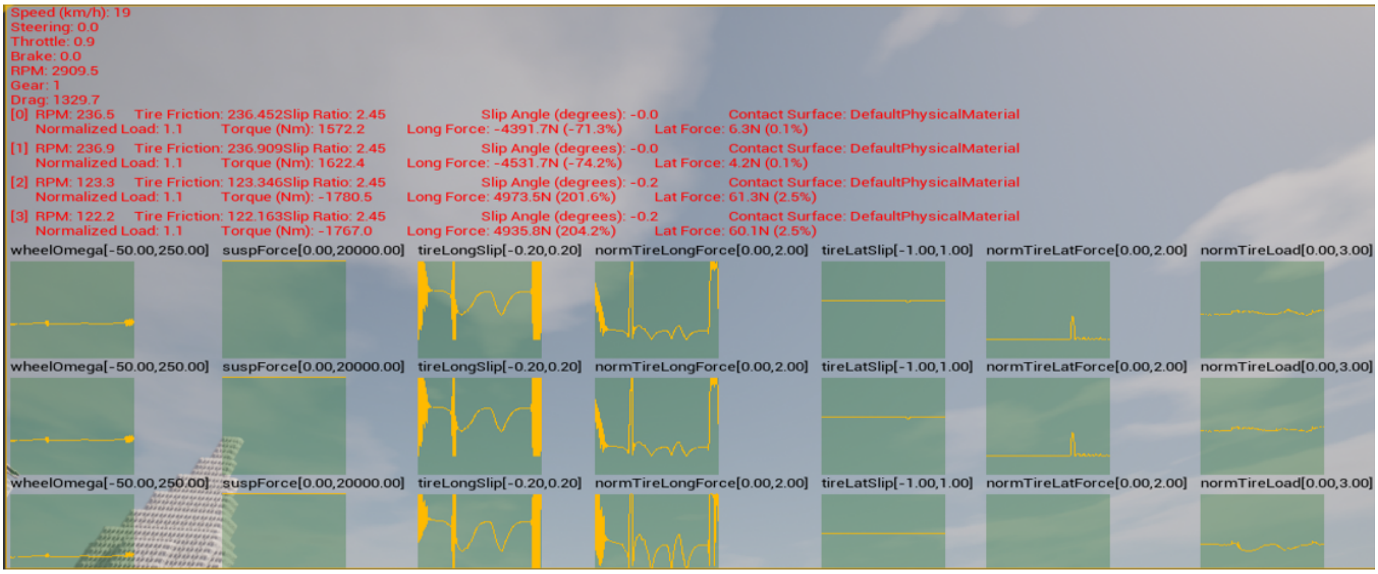

Fig. 4. Vehicle control input and feedback visualization.

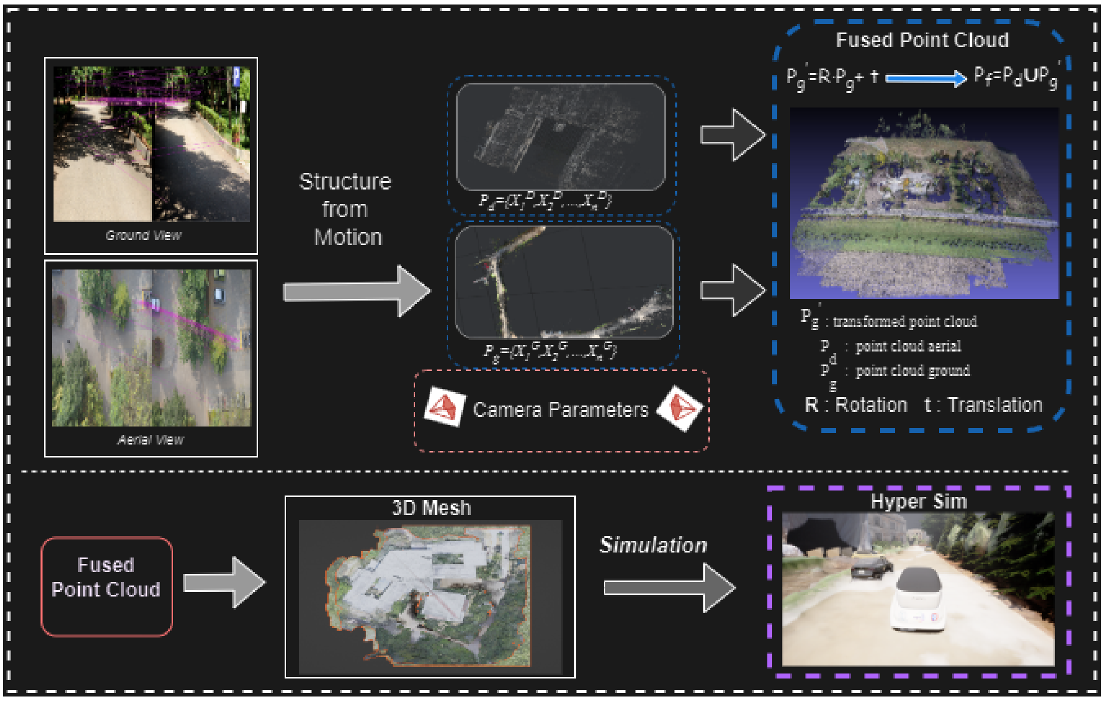

Fig. 5. Hyper Sim map generation pipeline. X_b^a represents a 3D point captured by drone or ground vehicle. To fuse the point clouds, the first step involves aligning them within a common coordinate system. This requires finding a rigid transformation consisting of rotation R and translation t to align P_g and P_d. After alignment, they are fused using Iterative Closest Point algorithm for point cloud alignment and fusion, denoted by P_f, to create a comprehensive mesh. The fusion process is expressed mathematically as: $P_d \cup P_{g'}$.

provide a comprehensive approach to reconstructing the 3D structure of a scene from multiple 2D images.

B. Simulation

We developed our HyperVL-Sim platform by leveraging an open-source autonomous vehicle (AV) simulator. Specifically, we integrate sensors such as LiDAR, cameras, and event sensors directly from the Unreal Engine 4 (Ue4) based CARLA simulator to enhance the simulation environment [1]. Following the mesh generation using Structure from Motion (SfM), the 3D mesh is directly imported into Unreal Engine 4, as seen in Fig. This approach eliminates the need for manually creating 3D maps, which is often time-consuming and less realistic. Our method of generating maps through 3D reconstruction provides enhanced performance for subsequent simulation tasks discussed in the upcoming section.

Method	FID	LPIPS
3DGS	182.82	0.7618
HyperVL-Sim (ours)	241.66	0.7515

TABLE I
MESH QUALITY ASSESSMENT. THE MESHES ARE EVALUATED USING THE SAME NUMBER OF IMAGES FOR BOTH METHODS.

C. Sensor Simulation

We build our sensor suit on top of unreal engine-based Carla simulator [1]. Using HyperVL-Sim we can simulate various high fidelity sensor data for autonomous driving autonomous driving application, as shown in Fig. 3. These sensors can be used to run high fidelity simulation as well as for generating synthetic data for Sim2Real domain adaptation. The sensors include Optical camera, event camera, LiDAR (Light Detection and Ranging) and Radar. In this work only the optical sensor has been used for all the experiments.

D. Vehicle in the Loop

In HyperVL-Sim, we introduce vehicle in loop simulation along with high fidelity hyper realistic simulation. HyperVL integrates a real autonomous vehicle system with a virtual clone simulated inside simulator, where virtual sensors provide data that is fed back to the vehicle's actuators, enabling closed-loop operation. The Vehicle-in-the-Loop (VIL) testing process involves several sequential steps as illustrated in Fig.6. The HyperVL-Sim is integrated with a physical autonomous vehicle in a test environment via a ROS node over a Wi-Fi network. Once the simulation begins, the command velocity, steering angle, and brake inputs are transmitted to the vehicle through a controller node, using ROS as middleware. The vehicle receives control inputs from the simulation and sends feedback to HyperVL-Sim as seen in Fig. 4 . The vehicle dynamics is modelled inside the HyperVL-Sim as seen in Fig.8 The command velocity is then used to derive control signals for the actuators based on the vehicle dynamics model within the simulator, completing the closed-loop operation. Additionally, any scenario can be simulated within the virtual HyperVL-Sim environment, allowing for comprehensive real-time actuation verification. Obstacles can be added at very fast speed and the actuation response time can be verified along with algorithm validation.

HyperVL-Sim thus enables the verification of control systems and chassis control systems through vehicle testing, reflecting the dynamic characteristics of actual vehicles along with algorithm validation and Sim2Real transfer learning.

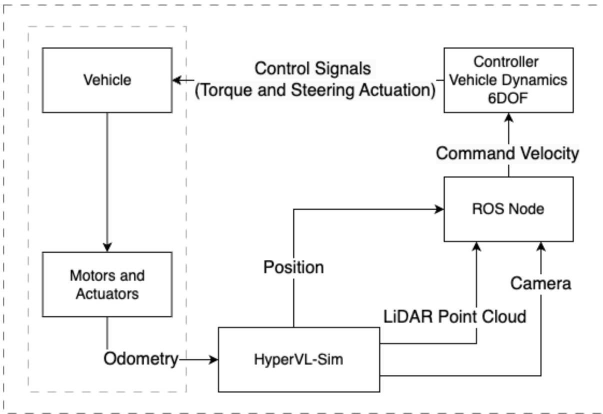

Fig. 6. Vehicle In Loop architecture.

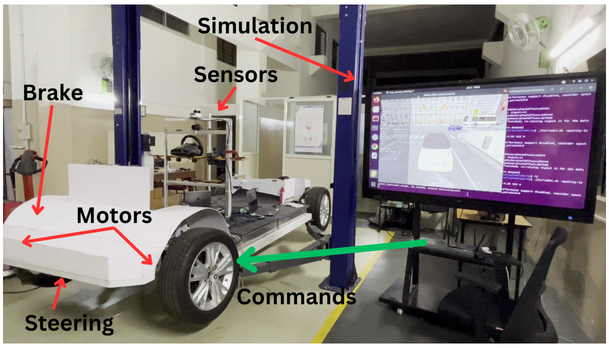

Fig. 7. Vehicle in the Loop testing apparatus.

IV. HARDWARE SETUP

The Vehicle In Loop setup consists of custom build autonomous SUV (Sport utility vehicle) system. The simulation platform consists of Intel i9-12900 Processor 24Gb NVIDIA RTX 3090 GPU. The models were trained on 24GB NVIDIA RTX 4090 GPU.

V. EXPERIMENTS

A. Vehicle in the Loop

HyperVL-Sim is validated for downstream tasks using Sim2Real transfer learning. To evaluate the performance of deep learning models on tasks such as object detection and semantic segmentation, the models are trained using data generated from HyperVL-Sim and are compared with models trained on data from CARLA and real-world datasets.

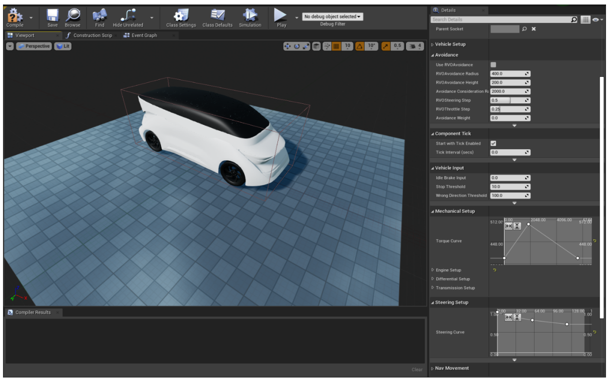

Fig. 8. Vehicle dynamics modelling using Ue4.

YOLO (You Only Look Once) network is selected for training deep learning models [29]. Unlike traditional models like MaskRCNN etc. that typically involve multiple stages, YOLO processes an entire image with a single neural network, making it highly efficient [30]. The model divides the image into a grid and predicts bounding boxes and class probabilities directly for each grid cell, allowing it to detect multiple objects in an image simultaneously. The experiments have been divided into two sections:

- Drivable Region Segmentation
- Object Detection

Object Detection

Three datasets were used: HyperSim M1, HyperSim M2, and a real-world dataset. The HyperSim datasets were generated from the maps depicted in Fig.1 and Fig.5, respectively. HyperSim M1 represents an off-road environment, while HyperSim M2 more closely resembles the real-world dataset, which was collected on campus and annotated. The CARLA dataset, specifically Town07, was selected as a synthetic benchmark due to its visual similarity to the real-world data. The dataset consists of around 350 images with, train:test:val split as 0.7:0.2:0.1. For testing purpose, all the models were validated on real world dataset. Only car class was selected for this experiment.

A YOLOv8 model was selected and trained for 100 epochs with a batch size of 1 on all three datasets. All models were pre-trained on MS COCO-128 dataset . The model performance can be seen in Table II. As seen from the Table, the model trained on Carla synthetic dataset fails to perform on the realdaset with $mAP50(MeanAveragePrecision)$ of 0.33 and $mAP50-95$ of 0.214. In comparison, HyperM2 map outperformed the carla dataset by almost doubling the $mAP50-95$ and other metrics. Despite significant visual disparities between the Hyper M1 map and the real dataset, Hyper M1 demonstrated comparable performance in object detection. The superior performance of Hyper M1 compared to the Carla dataset can be attributed to the higher fidelity of the HyperVL-Sim generated data. This increased realism enables models to generalize better to real-world scenarios than those trained solely on synthetic data like Carla.

[table,xcdraw]xcolor

Train On	Precision	Recall	% mAP50	% mAP50-95
Carla	0.521	0.373	0.335	0.214
Hyper M1	0.581	0.455	0.432	0.321
Hyper M2	0.740	0.679	0.720	0.583
Real	0.907	0.990	0.939	0.907

TABLE II
OBJECTION DETECTION PERFORMANCE COMPARISON.

Drivable Region

[table,xcdraw]xcolor

A YOLOv8 model was tuned for 50 epochs with 800 images from both datasets. We annotated about 400 images from the Carla and simulation out of which we used 100 images for Final validation the other 300 were augmented to generate another 800 images total. The tuning was started from the pre-trained weights for the MS-COCO dataset. We used another set of 100 images from real-world camera data to validate these tuned models.

Table IV presents performance metrics for lane detection models evaluated on two datasets: Simulation and Carla. Key metrics include Intersection over Union (IoU), F1 Score, Recall, and Precision. The goal is to generate a dataset that can be generalized to camera images.

[table,xcdraw]xcolor

Simulation Dataset: The model performs well, achieving a mIoU of 0.868, F1 Score of 0.892, Recall of 0.911, and Precision of 0.875. However, performance drops significantly when validated on the Carla dataset, with IoU at 0.421, F1 Score at 0.476, Recall at 0.547, and Precision at 0.426. The simulation dataset performance is significantly better on the camera dataset showing a clear difference in the features of the Carla and the camera dataset. This highlights the importance of training on data that closely resembles the target environment for effective detection.

Carla Dataset: When trained on Carla data, the model archives an IoU of 0.368, F1 Score of 0.384, Recall of 0.399, and Precision of 0.375. This indicates challenges in generalizing from simulated to real-world environments.

We see an improvement in the mIoU of 136% on the real-world camera data when the model is tuned using the hyper realistic simulation data. Indicating a clear performance improvement and better domain generalization.

B. Domain Generalization

It involves training models to generalize their learned features across various environments, making them robust to domain shifts and ensuring they work effectively in real-world applications. In this experiment, we used dataset of 300 images and only Hyper M2 map is used for training. Models were trained at 100 epoch with batch size as 1 and consistent learning rate. All the models were pre-trained on MS Coco 128 Dataset. As seen from the first row of Table III, the model

Experiment	Finetuned On	Pre-Trained	% mAP50 (Detect)	% mAP50-95 (Detect)	% mAP50 (Segment)	% mAP50-95 (Segment)
Sim2Real	Real	HyperVL-Sim	0.827	0.729	0.821	0.667
Sim2Real	HyperVL-Sim	Real	0.904	0.820	0.904	0.746
Real	Real	Coco 128	0.944	0.909	0.944	0.856

TABLE III
DETECTION AND SEGMENTATION PERFORMANCE FOR SIM2REAL DOMAIN GENERALIZATION. THE REAL DATASET WAS REDUCED BY 50% AND REPLACED BY THE HYPERVL-SIM DATASET.

Data Used to Train	Hyper VL			Carla		
Data Used to Validate	**Real**	**Hyper VL**	**Carla**	**Real**	**Hyper VL**	**Carla**
IOU	0.868	0.867	0.421	0.368	0.368	0.937
F1	0.892	0.883	0.476	0.384	0.384	0.967
Recall	0.911	0.890	0.547	0.399	0.399	0.980
Precision	0.875	0.877	0.426	0.375	0.375	0.955

TABLE IV
PERFORMANCE METRICS FOR DIFFERENT MODELS AND VALIDATION DATA

undergoes a two-stage training process. Initially the model is first pre-trained on HyperVL-Sim dataset and then fine tuned on real dataset. For this experiment we reduced the real dataset size by 50% to around 150 images and increase the HyperVL-Sim dataset size accordingly to have fixed dataset size. In the second row, the model was first trained on real dataset and then fine tuned to HyperVL-Sim dataset. The real dataset was finetuned on pre-trained MS COCO 128 dataset. In comparison with Table.II, the mAP50-90 for detection has increased by around 40% from 0.583 to 0.82. Notably, this model achieves performance comparable to a model trained exclusively on the real dataset, demonstrating the effectiveness of the HyperVL-Sim dataset in bridging the Sim2Real gap. Further, the model performs well on the segmentation task as well, achieving a mAP50-95 score of 0.746. Further, the dependence on the real data set has decreased by 50%, due to HyperVL-Sim dataset. This leads to lesser dependence on the real world dataset and helps increase the data generation time from hours to minutes.

VI. CONCLUSION

This research introduced HyperVL-Sim, a novel hyper-realistic Vehicle-in-Loop simulator designed to bridge the gap between simulation and reality in autonomous vehicle development. By integrating multiple sensor modalities and actual autonomous vehicle hardware, HyperVL-Sim creates a highly realistic simulation environment that mirrors real-world constraints, such as processing power and hardware latency. This hyper-realism is crucial for developing algorithms that are robust and reliable in safety-critical situations and also for dataset generation.

A key feature of HyperVL-Sim is its ability to generate high-fidelity maps using Structure from Motion (SfM) techniques. By fusing aerial and ground point clouds, HyperVL-Sim produces detailed and accurate maps that are essential for realistic simulation. These high-fidelity maps enable precise environmental modeling, which is critical for the development and testing of autonomous vehicle algorithms.

Our results demonstrate the effectiveness of HyperVL-Sim in accelerating Sim2Real transfer learning. With HyperVL-Sim we are able to achieve an mAP of 0.875 for drivable region segmentation—significantly outperforming CARLA's 0.375. Further, for object detection HyperVL-Sim, we were able to achieve an mAP of 0.583 as compared with 0.214 for Carla. For domain generalization, HyperVL-Sim was able to achieve 0.82mAP and 0.74mAP for combined detection and segmentation in domain adaptation tasks, while reducing the real dataset significantly by 50%.

HyperVL-Sim represents a critical advancement in autonomous vehicle simulation, providing a powerful tool for developing and testing the next generation of self-driving technology. Its ability to create a more accurate and reliable bridge between simulation and real-world conditions, supported by high-fidelity maps generated from fused aerial and ground point clouds, is essential for the continued progress toward safer autonomous vehicles.

REFERENCES

[1] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[2] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics: Results of the 11th International Conference*. Springer, 2018, pp. 621–635.

[3] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta *et al.*, "Lgsvl simulator: A high fidelity simulator for autonomous driving," in *2020 IEEE 23rd International conference on intelligent transportation systems (ITSC)*. IEEE, 2020, pp. 1–6.

[4] J. Shermeyer, T. Hossler, A. Van Etten, D. Hogan, R. Lewis, and D. Kim, "Rareplanes: Synthetic data takes flight," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 207–217.

[5] M. Deitke, W. Han, A. Herrasti, A. Kembhavi, E. Kolve, R. Mottaghi, J. Salvador, D. Schwenk, E. VanderBilt, M. Wallingford *et al.*, "Robothor: An open simulation-to-real embodied ai platform," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3164–3174.

[6] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: a survey," in *2020 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2020, pp. 737–744.

[7] Z. Szalay, "Next generation x-in-the-loop validation methodology for automated vehicle systems," *IEEE Access*, vol. 9, pp. 35 616–35 632, 2021.

[8] C. Brogle, C. Zhang, K. L. Lim, and T. Bräunl, "Hardware-in-the-loop autonomous driving simulation without real-time constraints," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 3, pp. 375–384, 2019.

[9] T. Tettamanti, M. Szalai, S. Vass, and V. Tihanyi, "Vehicle-in-the-loop test environment for autonomous driving with microscopic traffic simulation," in *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. IEEE, 2018, pp. 1–6.

[10] Aptiv, "Vehicle-in-the-loop-testing," 2022, https:https://www.aptiv.com/en/insights/article/what-is-vehicle-in-the-looptesting [Accessed: (2022)].

[11] Applus, "Vehicle-in-the-loop-testing," 2024, https://www.applusidiada.com/global/en/what-we-do/service-sheet/Vehicle-in-the-loop-testing [Accessed: (2024)].

[12] R. Jackson, A. Vamivakas, R. Jackson, and A. Vamivakas, "An overview of hardware-in-the-loop simulations for missiles," in *Modeling and Simulation Technologies Conference*, 1997, p. 3833.

[13] J. Noon, H. Song, B. Wen, R. Burgos, I. Cvetkovic, D. Boroyevich, S. Srdic, and G. Pammer, "A power hardware-in-the-loop testbench for aerospace applications," in *2020 IEEE Applied Power Electronics Conference and Exposition (APEC)*. IEEE, 2020, pp. 2884–2891.

[14] P. D. Achlerkar and B. K. Panigrahi, "Robust backstepping output voltage controller for standalone voltage-sourced converters," *IET Renewable Power Generation*, vol. 14, no. 12, pp. 2211–2220, 2020.

[15] A. S. Abdelrahman, K. S. Algarny, and M. Z. Youssef, "A novel platform for powertrain modeling of electric cars with experimental validation using real-time hardware in the loop (hil): A case study of gm second generation chevrolet volt," *IEEE Transactions on Power Electronics*, vol. 33, no. 11, pp. 9762–9771, 2018.

[16] L. G. G. Soeiro and B. J. C. Filho, "Vehicle power system modeling and integration in hardware-in-the-loop (hil) simulations," *Machines*, vol. 11, no. 6, p. 605, 2023.

[17] H. K. Fathy, Z. S. Filipi, J. Hagena, and J. L. Stein, "Review of hardware-in-the-loop simulation and its prospects in the automotive area," in *Modeling and simulation for military applications*, vol. 6228. SPIE, 2006, pp. 117–136.

[18] W. Deng, Y. H. Lee, and A. Zhao, "Hardware-in-the-loop simulation for autonomous driving," in *2008 34th Annual Conference of IEEE Industrial Electronics*. IEEE, 2008, pp. 1742–1747.

[19] T. Qin, C. Li, H. Ye, S. Wan, M. Li, H. Liu, and M. Yang, "Crowd-sourced nerf: Collecting data from production vehicles for 3d street view reconstruction," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[20] L. He, L. Li, W. Sun, Z. Han, Y. Liu, S. Zheng, J. Wang, and K. Li, "Neural radiance field in autonomous driving: A survey," *arXiv preprint arXiv:2404.13816*, 2024.

[21] X. Hu, S. Li, T. Huang, B. Tang, R. Huai, and L. Chen, "How simulation helps autonomous driving: A survey of sim2real, digital twins, and parallel intelligence," *IEEE Transactions on Intelligent Vehicles*, 2023.

[22] J. Revell, D. Welch, and J. Hereford, "Sim2real: Issues in transferring autonomous driving model from simulation to real world," in *SoutheastCon 2022*. IEEE, 2022, pp. 296–301.

[23] W. Li, C. Pan, R. Zhang, J. Ren, Y. Ma, J. Fang, F. Yan, Q. Geng, X. Huang, H. Gong *et al.*, "Aads: Augmented autonomous driving simulation using data-driven algorithms," *Science robotics*, vol. 4, no. 28, p. eaaw0863, 2019.

[24] Z. Yang, Y. Chen, J. Wang, S. Manivasagam, W.-C. Ma, A. J. Yang, and R. Urtasun, "Unisim: A neural closed-loop sensor simulator," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1389–1399.

[25] M. J. Westoby, J. Brasington, N. F. Glasser, M. J. Hambrey, and J. M. Reynolds, "'structure-from-motion'photogrammetry: A low-cost, effective tool for geoscience applications," *Geomorphology*, vol. 179, pp. 300–314, 2012.

[26] C. Griwodz, S. Gasparini, L. Calvet, P. Gurdjos, F. Castan, B. Maujean, G. De Lillo, and Y. Lanthony, "Alicevision meshroom: An open-source 3d reconstruction pipeline," in *Proceedings of the 12th ACM multimedia systems conference*, 2021, pp. 241–247.

[27] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[28] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[30] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.