Training and Evaluating Machine Learning Models using XR Simulated Data for Autonomous Vehicle Control

Adam Kohl, Eliot Winer
Iowa State University
Ames, IA
adamkohl@iastate.edu, ewiner@iastate.edu

ABSTRACT

The Department of Defense (DOD) increased funding for Artificial Intelligence (AI) from $874 million in 2022 to approximately $1.8 billion in 2024. The allocation demand signifies a strategic push for machine learning (ML) driven capabilities, which includes Autonomous Vehicle (AV) control and navigation in a variety of military applications. However, physical tests of AVs incur high development costs and introduces substantial safety risks. Extended Reality (XR) simulation platforms provide a safe and cost-effective alternative to determine viability and repeat testing but may significantly differ from real-world driving. If XR could be proven as a viable platform for traffic data collection, driving scenarios, such as AVs, could be studied extensively. This research described in this paper studies the use of a custom multimodal traffic simulation platform (i.e., InterchangeSE) to simulate various driving scenarios and then use the resulting data to train a ML AV algorithm. InterchangeSE possesses four key capabilities: autonomous vehicle integration, 3D virtual environment (VE) rendering, human agent interfacing, and traffic generation.

This paper presents the training, deployment, and evaluation of various ML models using data from InterchangeSE. Two scenarios, a baseline road environment with and without traffic, were used for data collection. Using Design of Experiments (DOEs), three ML end-to-end learning architectures were subjected to variability in their network design and common image augmentation techniques, such as horizontal flipping, brightness variability, additional noise, and shadow manipulation. Alongside investigating model training, a detailed statistical analysis was performed to assess the impacts of these model configurations under the real-time constraints imposed by the virtual environment. The top performing models were deployed and evaluated in the XR simulation platform under different driving scenarios, including three-lane highways, highway merging, and intersections, with the findings yielding effective AV control.

ABOUT THE AUTHORS

Adam Kohl is a Ph.D. candidate in Mechanical Engineering and Computer Engineering at Iowa State University's VRAC Research center. His research focuses on advancing deep learning techniques to optimize model predictive control and simulate human behavior in complex systems.

Eliot Winer is the director of the VRAC Research Center and professor of Mechanical Engineering, Electrical and Computer Engineering, and Aerospace Engineering at Iowa State University. Dr. Winer has over 25 years of experience working in extended reality and 3D computer graphics technologies on sponsored projects for the Department of Defense, Air Force Office of Scientific Research, Department of the Army, National Science Foundation, Department of Agriculture, Boeing, John Deere, and the Federal Highway Administration.

Training and Evaluating Machine Learning Models using XR Simulated Data for Autonomous Vehicle Control

Adam Kohl, Eliot Winer
Iowa State University
Ames, IA
adamkohl@iastate.edu, ewiner@iastate.edu

INTRODUCTION

The pursuit of autonomous vehicle (AV) technology has evolved from an ambitious research challenge into a transformative industry, which has profound implications for both civilian and military applications. In 2004, the Defense Advanced Research Projects Agency (DARPA) catalyzed this evolution through its $1 million Grand Challenge, which tasked competing teams with developing a driverless vehicle capable of autonomously navigating 132 miles of Mojave Desert terrain within a 10-hour timeframe (Behringer et al., 2004). Although no team completed the course, Carnegie Mellon University's vehicle traveled 7.3 miles, marking a significant milestone that ignited global interest in AV development (Buehler et al., 2007). This seminal event laid the groundwork for an industry that has since expanded rapidly, with the global AV market projected to reach $273.75 billion by 2025 (Precedence Research, 2025). Major investments, including Waymo's $5.6 billion funding round in 2024 (O'Kane, 2024) and Tesla's $10 billion commitment to AV technology (Alvarez, 2024), underscore the strong commercial interest in advancing AV capabilities. In the civilian sector, AV technologies promise improved road safety, reduced traffic congestion, lower emissions, and decreased driver fatigue, addressing critical societal challenges, such as the 1.35 million annual traffic fatalities reported globally (World Health Organization, 2023).

Parallel to civilian advancements, the Department of Defense (DoD) has recognized the strategic importance of AVs for enhancing military operations, particularly in high-risk logistics scenarios. To address challenges such as these, the DoD has significantly increased its investment in artificial intelligence (AI) and autonomous systems. Specifically, funding rose from $874 million in 2022 to $1.8 billion in 2024, signaling strong support for machine learning (ML) driven solutions, such as AV control and navigation (Gray, 2025; U.S. Department of Defense, 2023). In 2024, the DoD also allocated $10.95 billion for uncrewed vehicle programs, advancing autonomous technologies critical for military applications (AUVSI, 2024). These financial commitments underscore the DoD's strategic prioritization of integrating AI and autonomy to enhance military capabilities in various operational scenarios. For example, during operations in Iraq and Afghanistan, improvised explosive devices (IEDs) posed substantial threats to convoy personnel, with over 3,500 U.S. military fatalities attributed to such attacks between 2001 and 2012 (Cordesman & Lin, 2015). It would now be in the realm of possibility to have AVs in many of these situations using end-to-end supervised learning. These AVs, effectively ML agents, could be trained on human-like driving behavior data captured from simulations. The AVs would then replace humans, where possible, removing them from exposure to IEDs and other dangers in hazardous operating environments.

The development and validation of AVs for military applications face significant challenges, including high costs and safety risks associated with physical testing. Real-world testing of AVs requires extensive resources, dedicated proving grounds, and stringent safety protocols, often rendering it impractical for iterative design cycles (National Highway Traffic Safety Administration, 2022). The complexity of military operating environments, ranging from urban terrains to unstructured off-road settings, demands robust validation of AV control and navigation systems under diverse conditions (U.S. Army, 2021). To address this challenge, XR simulation platforms offer a safe and cost-effective means to replicate these conditions and collect critical data for training ML models. These environments offer the extensibility to be fully populated with computer-controlled agents (e.g., drivers and traffic) or a hybrid of humans and computer-controlled agents. In addition, a well-constructed XR environment would allow for the integration of multiple modalities of agents such as overall traffic, AVs, pedestrians, etc. allowing for a much more realistic environment from which to capture data. The work presented in this paper evaluates the performance of supervised ML architectures for AV control within the multimodal traffic simulation platform InterchangeSE (Miller et al., 2022). The research examines the influence of ML model architectures, training datasets, diverse driving scenarios, and two distinct simulation environments on AV performance. The investigation addresses several key research questions, including:

- To what extent can a ML model replicate human driving behavior across varied driving scenarios using only collected data from a VE?
- How does data augmentation affect the performance of the ML model in a VE?
- How effectively can an AV, trained solely on a dataset without traffic, perform in a traffic-inclusive VE?

The following section provides background on simulation platforms for AV research, covering individual vehicle and traffic simulation, their integration in XR, and supervised end-to-end ML, with a focus on Convolutional Neural Networks (CNNs) for predictive control. The methodology will then be discussed, including the InterchangeSE framework architecture, simulated driving environments, ML architectures (i.e., PilotNet, Branched PilotNet, and JNet), hardware utilization, and model deployment within the simulation. The remaining sections will present results of the simulations run and ML models created.

BACKGROUND

Simulation Platforms for AV Research Development

The development of autonomous vehicle control systems hinges on the use of simulation platforms, which provide safe, controlled, and cost-effective environments for testing and refining driving algorithms. These platforms differ widely in their fidelity, focus, and capabilities, addressing various facets of autonomous vehicle research, including perception, control, and traffic management. A few examples of AV algorithms are described in the paragraphs below that provide good representations of what is available in the commercial and open-source marketplace. This is followed by a brief section on supervised end-to-end Learning

Built on Unreal Engine, CARLA (Dosovitskiy et al., 2017) operates using a client-server architecture: the server manages the simulation (e.g., handling physics, rendering, and world dynamics), while clients interact with it through a Python or C++ API to control actors, adjust settings, and collect data. The core focus of CARLA is to simulate realistic driving scenarios to test machine learning models, such as perception algorithms or driving policies, and offers highly realistic urban environments with detailed road networks, buildings, pedestrians, and dynamic weather conditions (Dosovitskiy et al., 2017). A key strength lies in its ability to simulate complex traffic scenarios and multi-agent interactions. Yet, its computational intensity can limit scalability for large-scale simulations (i.e., hundreds of vehicles and expansive spatial regions), and its predefined scenarios often require customization to address rare edge cases effectively.

NVIDIA DRIVE Sim is a comprehensive simulation platform designed to address one of the most significant challenges in AV development: the need for vast amounts of diverse, high-quality data to train and test self-driving systems (NVIDIA, 2024).This platform takes a different approach, emphasizing photorealistic rendering and precise physics modeling to simulate vehicle dynamics and sensor performance. Built on advanced ray-tracing technology, this closed-source platform excels in simulating sensor outputs, such as those from cameras and LiDAR, making it an excellent choice for developing and testing perception systems under diverse environmental conditions. However, its reliance on high-performance NVIDIA GPUs drives up costs and restricts accessibility, particularly for smaller research groups. Additionally, DRIVE Sim prioritizes visual and physical fidelity over the simulation of complex human behaviors, such as pedestrian or cyclist movements, which are essential for holistic autonomous vehicle evaluation.

For researchers focused on vehicle control and reinforcement learning, TORCS (The Open Racing Car Simulator) offers a lightweight, open-source alternative (Espié et al., 2005). Originally developed for racing simulations, TORCS provides customizable tracks and vehicle models, making it ideal for training reinforcement learning agents on tasks like lane-keeping and overtaking. Its simplicity and compatibility with machine learning frameworks enhance accessibility, though its simplified physics and lack of advanced sensor support limit its applicability to real-world AV scenarios requiring high fidelity. Unfortunately, CARLA, NVIDIA DRIVE Sim, and TORCS do not natively provide VR, XR, or AR environments as part of their out-of-the-box features. Their default setup is geared toward desktop simulation, not immersive technologies. However, a platform like CARLA is built on Unreal Engine, which has robust support for VR and AR development. It is technically possible to produce an XR environment, but at a large cost of additional development effort with limited capabilities.

Transitioning from individual vehicle simulation to traffic dynamics, Simulation of Urban Mobility (SUMO) excels in modeling large-scale traffic networks (Lopez et al., 2018). Using an extension of the Gipps' model, SUMO simulates traffic patterns across diverse scales, from intersections to city-wide systems, incorporating vehicles, pedestrians, and public transit. Its primary strength lies in analyzing traffic flow and testing higher-level control systems, such as traffic-aware path planning. However, its lack of sensor simulation and low graphical fidelity constrain its utility for perception or detailed vehicle control studies. Complementing SUMO, Vissim is a commercial microscopic traffic simulator developed by PTV Group, tailored for traffic engineering and transportation planning (Fellendorf & Vortisch, 2010). It models individual vehicle and pedestrian behaviors with high granularity, offering detailed insights into traffic dynamics. Its ability to exchange data via shared memory supports integration with other simulators, enhancing its role in multi-fidelity setups. However, Vissim's limited visual realism and absence of sensor simulation reduce its applicability for perception-centric AV research.

Supervised End-to-End Learning

Supervised end-to-end machine learning is a significant departure from the traditional supervised ML pipelines by bypassing the intermediate feature engineering and preprocessing steps, which consume a considerable number of resources. The introduction of supervised end-to-end learning for autonomous control can be traced back to 1988 with Pomerleau's ALVINN, a seminal neural network designed to predict steering angles from camera inputs for basic lane-following tasks (Pomerleau, 1988). Though limited in scope and computational power, ALVINN established a foundational proof-of-concept that continues to inspire subsequent research. The field gained significant momentum with the rise of convolutional neural networks (CNNs) and the proliferation of large-scale datasets, which together enabled the development of more robust and versatile architectures.

A landmark in this progression is NVIDIA's PilotNet, introduced by Bojarski et al., which processes 200x66 pixel images through a streamlined Convolutional Neural Network (CNN) to predict steering angles, which yielded a Mean Squared Error (MSE) of 0.005 when trained and exposed to collected real-world images (Bojarski et al., 2016). Demonstrating practical efficacy on highways, PilotNet set a benchmark for modern end-to-end models. This framework was later extended by Xu et al., with the PilotNet Branched Fully Connected Network (FCN), a fully convolutional neural network that simultaneously predicts steering angles and acceleration, achieving MSEs of 0.004 and 0.002, respectively, in simulated environments, thus broadening the scope of control outputs.

As the research field matured, addressing interpretability became a priority, particularly given the safety-critical nature of autonomous driving. Kim and Canny introduced JNet, which incorporates attention mechanisms to highlight salient image features such as road edges, reducing steering error by 15 percent compared to PilotNet, while offering valuable insights into the model's decision-making process. Earlier, Chen et al. contributed DeepDrive, a CNN-based model that pioneered direct perception approaches with an MSE of 0.008 for steering prediction, influencing subsequent designs. Industry efforts have also played a pivotal role, with Intel's DriveNet building on PilotNet's foundation to enhance robustness across diverse conditions, achieving an MSE of 0.0045 in real-world tests. These advancements reflect a growing emphasis on practical deployment and reliability.

The evolution of supervised end-to-end machine learning for AV control has progressed significantly from Pomerleau's foundational ALVINN to sophisticated models like PilotNet, PilotNet Branched and JNet, which leverage CNNs and attention mechanisms to achieve low mean squared errors and enhanced interpretability. These advancements underscore the potential for robust, real-time autonomous control while addressing critical safety and reliability concerns. For this research, three architectures were selected: NVIDIA's PilotNet, PilotNet Branched, and JNet. PilotNet was selected as the primary baseline architecture due to its proven efficacy in highway scenarios, streamlined design, and established benchmark performance (Bojarski et al., 2016). PilotNet Branched (Xu et al., 2017) extends PilotNet with a fully convolutional network to predict multiple control outputs, enhancing its versatility for complex driving tasks. JNet (Kim and Canny, 2017) incorporates attention mechanisms for improved interpretability and features a smaller architecture with a reduced parameter count, enabling efficient computation while maintaining robust performance. These architectures were chosen for their complementary strengths: PilotNet's established benchmark performance, PilotNet Branched's multi-output capability, and JNet's compact, interpretable design. These characteristics make the three architectures well-suited for comprehensive evaluation within the InterchangeSE XR simulation platform.

METHODOLOGY

Multimodal Simulation Environment

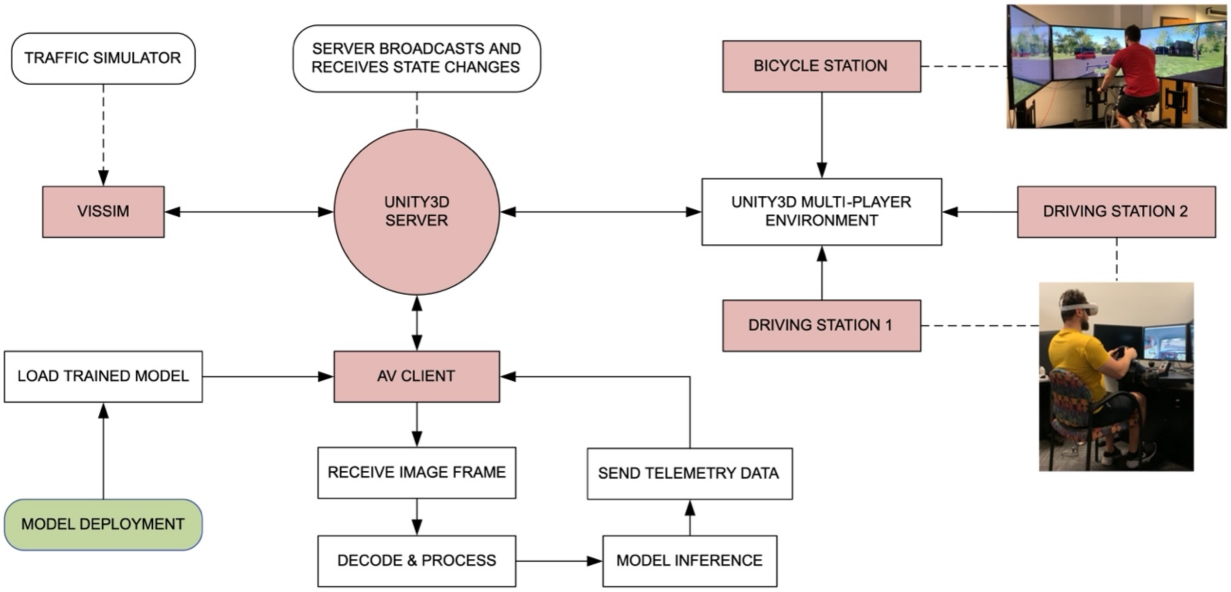


Figure 1. InterchangeSE Multimodal Traffic Simulation Framework Architecture

The system architecture for InterchangeSE is shown in Figure 1. The system architecture, as depicted, consists of three core components: 1) a traffic simulation module, 2) a server enabling bidirectional communication of state changes, and 3) clients representing physical vehicle rigs and autonomous vehicles (AVs). The client-server framework was implemented using the Unity game engine (Unity Technologies, 2005), which also functions as the visualization platform for desktop and virtual reality environments. For a given driving scenario, such as an urban neighborhood or a highway, a corresponding computational road network was developed in VISSIM. This network was employed to simulate vehicle and pedestrian traffic entities within VISSIM. The state data of each vehicle entity were transmitted to the Unity server, which then broadcasted these updates to all connected clients. Each client maintains self-awareness of its position and orientation, allowing, for example, a cyclist to track the position of an AV within the scenario. The driving states of clients, encompassing position and orientation, are sent to VISSIM via the Unity server at every frame. These client states are subsequently processed and integrated as new entities within VISSIM.

To deploy a trained ML model for AV control, the model operates as a separate process and establishes a connection to the Unity server. Configured with predefined parameters, the trained AV model is loaded and awaits the broadcast of image data from the server. Upon receipt, the image data are decoded and preprocessed to meet the input requirements of the model. The model then processes the prepared image data to generate predicted outputs, specifically steering angle and throttle response values, which are transmitted back to the Unity server for integration into the simulation.

ML Architectures

The first adapted architecture was PilotNet, a foundational end-to-end CNN, as described earlier. In the context of the InterchangeSE simulation platform, PilotNet processes RGB images from a front-facing virtual camera to predict steering angle and throttle response. To enhance robustness against lighting variations common in virtual environments, a normalization layer is applied to the input images, which standardizes pixel intensities. The network's backbone is comprised of five convolutional layers with varying filter sizes: 1) larger 5x5 kernels in early layers to capture coarse features like road edges and 2) smaller 3x3 kernels in later layers to extract finer details such as lane markings. Each convolutional layer is followed by rectified linear (ReLU) activations to introduce non-linearity, enhancing the model's ability to learn complex feature representations. The convolutional stack is followed by three fully connected layers comprised of 100, 50, and 10 neurons with ReLU activations introducing non-linearity and followed with outputs of steering angle and throttle response.

The second adapted architecture was PilotNetBranched, which extends the original PilotNet to support multi-task learning, enabling simultaneous prediction of steering angle and throttle response through task-specific branches. Like PilotNet, it processes RGB images from a front-facing camera in the InterchangeSE platform. The architecture retains PilotNet's convolutional backbone to extract shared visual features, such as road geometry, traffic patterns, and lane boundaries before splitting into separate branches of fully connected layers for each output. One of the branches is tasked with predicting steering angle and the another for predicting throttle response. Each branch consists of three fully connected layers comprised of 100, 50, and 10 neurons with ReLU activations. However, the additional task-specific branch introduces more parameters, which increases the computational footprint and may strain real-time constraints in resource-limited XR simulations.

The final architecture adapted, JNet, is a lightweight CNN architecture ideal for resource-constrained environments. The network is designed to reduce the computational footprint of end-to-end autonomous driving models while maintaining performance comparable to larger models like PilotNet (Kocić et al., 2019). Similar to PilotNet, JNet process RGB images from a font-facing camera, and its lightweight design is achieved by reducing the number of convolutional layers, using smaller filter sizes, and minimizing parameters. For example, the number of parameters for the adopted JNet architecture total approximately 85k compared to PilotNet with approximately 252k and Branched PilotNet with approximately 373k. However, it can yield a less consistent performance in "challenging" scenarios. For example, due to the architecture's reduced capacity, navigating congested three-lane highways with frequent lane changes may cause the model to struggle to consistently predict optimal steering angles due to its simplified feature extraction. Compared to PilotNet, its simplified feature extraction may fail to robustly detect critical features like lane markings when introduced to noise or shadow effects in the simulated environment.

Data Collection

A dataset was created to train the aforementioned ML models, using InterchangeSE. First, a roadway was developed as shown in Figure 2 and put into InterchangeSE with a driving station as shown in the right of Figure 1. A user would begin by driving a vehicle in the environment using a gaming type steering wheel and appropriate gas and brake pedals. The user would then begin capturing the data by recording their input into the system. Recording the driving experience would produce a series of image and telemetry data as an attribute log file. Attributes were comprised of the image (i.e., the file path of the stored image), the steering angle, throttle response, and speed of the vehicle.

The driving environment shown in Figure 2 provides several different scenarios a driver would reasonably encounter. These scenarios serve as the trials to capture the needed image and telemetry data with which to train the ML model. Embodying the end-to-end supervised learning paradigm, the data collected will not be uniform in nature. For example, consider a scenario where a driver is told to get from one point on the three-lane highway and "circle" around the town to a point approximately 180 degrees away from it. One potential path would require the amount of time spent (i.e., data recorded) on the three-lane highway much greater than that spent merging on or off. Another route may require more four-way intersections compared to time on the three-lane highway. This variability is needed, as it represents realistic scenarios, but may prove troublesome when training the ML models. For data collection, drivers were instructed to navigate the InterchangeSE driving environment, completing a series of predefined routes that included traveling between specified points (e.g., from a starting location on the three-lane highway to a designated point in the residential sector) and performing general exploration of the environment to capture diverse driving behaviors. Following all the data capture it was observed that users spent approximately 50 percent of their time on the three-lane highway, 30 percent merging from the residential sector on the highway or merging off the highway into the residential sector, and the remaining 20 percent in the residential sector. To generate the data collected for training, approximately eleven 30-minute trials were carried out using a driving station as shown in Figure 1.

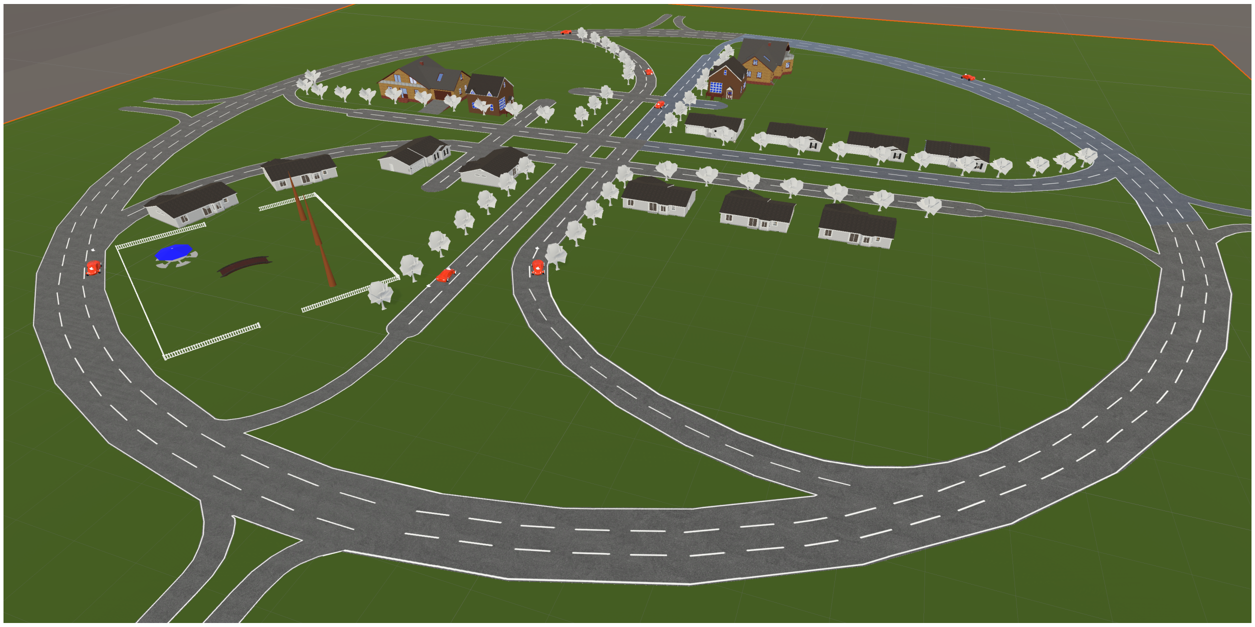


Figure 2. Driving environment used in the data collection process.

A total of 303,980 samples were collected to comprise the dataset: 146,818 belonging to driving on the three-lane highway, 98,677 merging on or off the highway, and 58,485 in the residential sector. Each sample contains an input variable (i.e., center image), two target variables (i.e., steering angle and throttle response), and a conditional variable (i.e., velocity) providing contextual information regarding the driving conditions. As seen in the left of Figure 3, steering angle is normalized from negative one to one and is highly skewed towards zero with a mean value of negative 0.3 and standard deviation of 0.18 degrees. The significant number of samples near zero degrees indicates the vehicle's collected trajectory is biased towards low or minimal steering angle adjustments. In other words, during data collection, most of the driving was relatively straight with few turns needed. Tails of the distribution, which represent large positive and negative steering angles, are of a much lower frequency implying a relatively rare occurrence of sharp turns. The distribution of the throttle response is shown in the right of Figure 3. The values range from zero to one, with a mean value of 0.56 and standard deviation of 0.36. The significant number of samples at the far edges of the spectrum suggest differing driving behaviors. For example, Figure 4 highlights the disparity amongst throttle response, where the residential sector contains a vast majority of values at zero and the three-lane highway contains many values at the maximum of one. These distributions illustrate the vehicle, from the collected data, isn't accelerating the majority of the time when in the residential sector and accelerates heavily when driving on the three-lane highway. InterchangeSE has a built-in governor that sets the maximum speed a user can drive in the simulation, which caused the throttle response to contain a large number of samples to be valued at one (i.e., the user had the pedal fully depressed but couldn't go beyond the maximum allowable speed). Overall, the throttle response distribution captures a mixture of stop-and-go traffic, cruising, and acceleration states, which appear to be reasonable for representing typical driving behaviors in residential and highway environments.

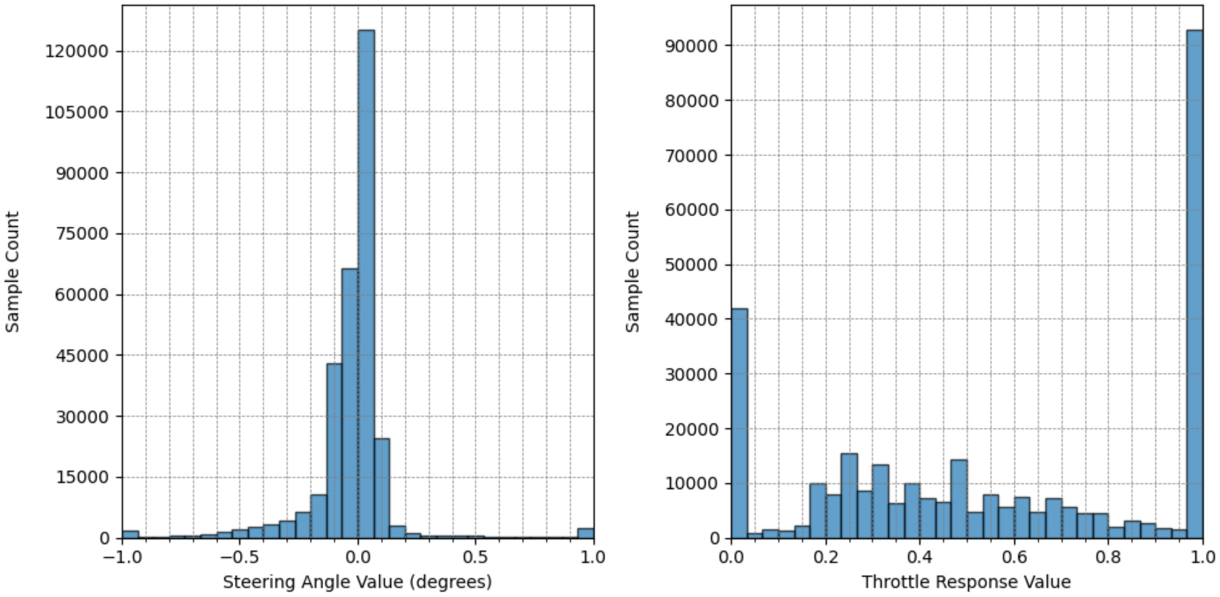


Figure 3. Collected data distribution of the steering angle and throttle response

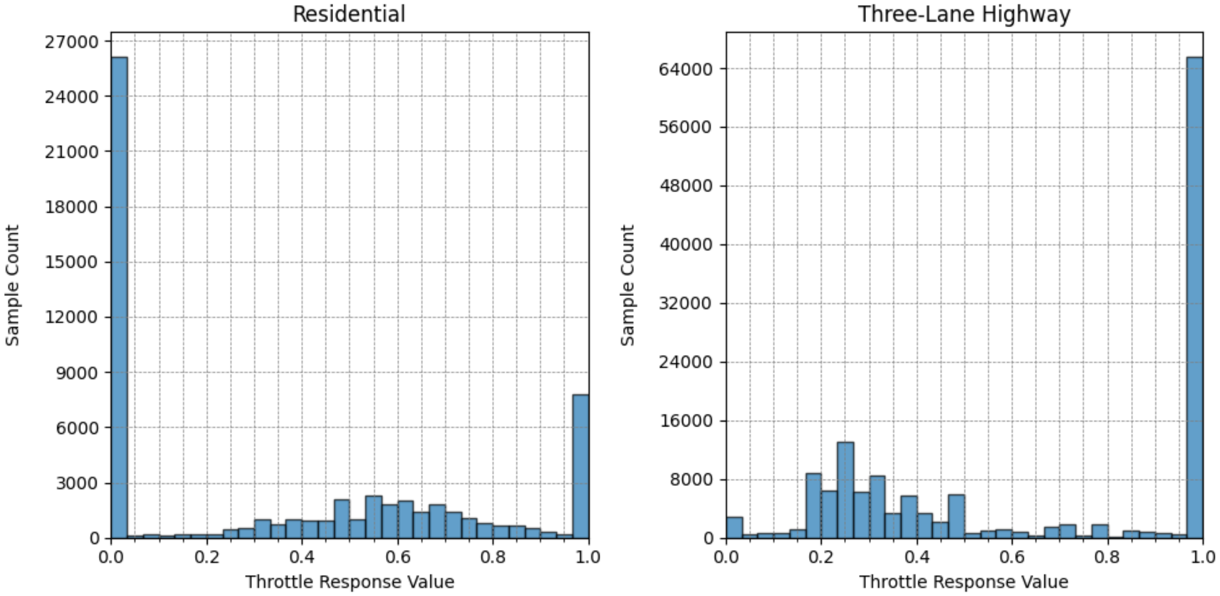
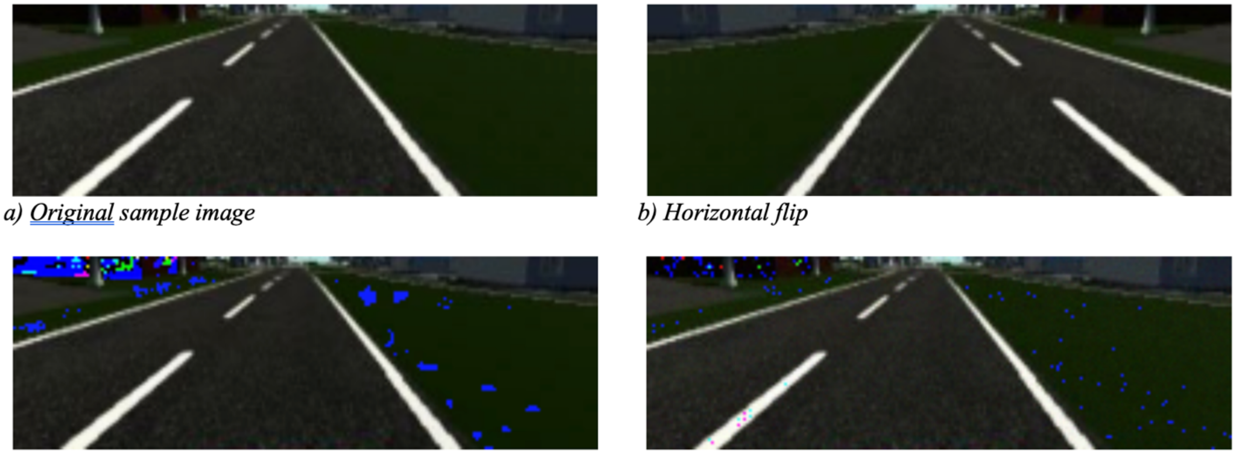


Figure 4. Throttle response distribution from the residential sector vs the three-lane highway

Training

Each of the three model architectures, PilotNet, Branched Pilot, and JNet were trained on the collected data to establish a baseline. Architecture variations including the use of a batch normalization layer and dropout layers after each fully connected layer were run to identify the best performing configuration for each model architecture. Using each model's best configuration, additional experiments were conducted on each of the architectures to see how data augmentation techniques, when applied during training, would affect the model performance. Images were augmented on 30 percent of the data at random intervals during each training epoch. Image augmentation is a widely adopted strategy in training ML models, particularly for computer vision tasks like AV control, to enhance model robustness, generalization, and performance under diverse real-world conditions (Mikołajczyk & Grochowski, 2018; Shorten & Khoshgoftaar, 2019). The selected augmentation techniques address specific challenges in AV perception and navigation, such as varying environmental conditions, sensor noise, and scene variability. The data augmentation consists of four techniques: 1) an image horizontal flip, 2) image brightness variation, 3) increased image gaussian noise variability, and 4) image shadow manipulation. Examples of these augmentation techniques applied to a sample of the dataset are shown in Figure 5.



a) Original sample image

b) Horizontal flip

c) Brightness variation

d) Gaussian noise variability

Figure 5. Data augmentation techniques applied to images during training

Horizontal flipping simulates mirrored road perspectives (e.g., opposite driving directions), enhancing model generalization across symmetric road geometries like turns or lanes in InterchangeSE's scenarios (e.g., highways, intersections). It ensures end-to-end models like PilotNet learn invariant features for bidirectional navigation (Bojarski et al., 2016) Brightness variation mimics lighting changes (e.g., day, night, weather), critical for robust feature extraction in varied illumination, which prepares CNNs to handle pixel intensity shifts in 3D environments (Shorten & Khoshgoftaar, 2019). Gaussian noise simulates sensor imperfections (e.g., camera noise in low light), improving model robustness to degraded image quality, which helps the model generalize to noisy real-world conditions, such as highway merging, absent in clean XR simulations (Mikołajczyk & Grochowski, 2018). Shadow manipulation replicates occlusions from objects (e.g., trees, buildings), teaching models to adapt to obscured road features in scenarios like intersections. It enhances reliability for complex scene processing in end-to-end architectures (Cubuk et al., 2019). Image augmentation was applied to 30% of the training data at random intervals during each epoch, a proportion consistent with common practices in computer vision research to balance variability and data fidelity (Cubuk et al., 2019; Shorten & Khoshgoftaar, 2019). This percentage was selected to introduce sufficient environmental variability, such as lighting changes and sensor noise, to enhance model robustness for AV control within the InterchangeSE platform, while preserving the majority of the original dataset's characteristics to reflect human driving behavior in the simulated scenarios.

RESULTS AND DISCUSSION

The evaluation of three ML architectures (i.e., PilotNet, PilotNet Branched, and JNet) within the InterchangeSE XR simulation platform yielded insights into their efficacy for autonomous vehicle (AV) control. The models were trained on a dataset comprised of 303,980 samples, with 146,818 samples from three-lane highway scenarios, 98,677 from highway merging, and 58,485 from residential sectors. Each model was assessed using Mean Squared Error (MSE) for steering angle and throttle response, alongside Huber loss to evaluate robustness against outliers. The baseline performance of the models, without data augmentation, is summarized in Table 1.

Table 1. Model architecture baseline evaluation

Model Architecture	Steering Angle MSE	Throttle Response MSE	# Epochs
PilotNet	0.00092	0.0319	70
PilotNet Branched	0.00065	0.0322	60
JNet	0.00112	0.0394	90

PilotNet Branched achieved the lowest steering angle MSE (i.e., 0.00065), indicating superior precision likely due to its multi-task learning capability with task-specific branches. An MSE of 0.00065 yields a root mean squared error (RMSE) of approximately 0.0255, which, when divided by the normalized steering range of 2 (i.e., [-1,1]) and multiplied by 100, indicates steering predictions deviate by about 0.8 percent from ground truth steering commands in the dataset on average, adjusted conservatively to reflect the data's skew toward small adjustments. This minimal deviation, equivalent to less than one degree off-target in a typical ±45-degree steering range, ensures precise control in complex scenarios like highway merging, comparable to a human driver making near-perfect corrections. PilotNet followed with a steering angle MSE of 0.00092, corresponding to an approximate 1.0 percent deviation, consistent with its established benchmark performance in highway scenarios. JNet, with a higher MSE of 0.00112, exhibited reduced accuracy (i.e., roughly a 1.1 percent deviation) attributable to its lightweight design with fewer parameters, which limits its capacity to capture complex features. For throttle response, PilotNet achieved the lowest MSE (i.e., 0.0319), while JNet's higher MSE (i.e., 0.0394) suggests challenges in modeling acceleration behaviors, particularly in varied scenarios like residential sectors with stop-and-go traffic.

The effects of the data augmentation techniques applied during model training on steering angle and throttle response were analyzed across the three architectures. Table 2 summarizes the best-performing models for each augmentation type with respect to Huber loss, which is a metric less sensitive to outliers. Horizontal flipping, simulating mirrored road perspectives, consistently degraded performance across all architectures, with PilotNet Branched yielding the lowest steering angle Huber loss of 0.00113 among the augmented models. This suggests that while flipping enhances generalization for symmetric road geometries, it may introduce confusion in the InterchangeSE environment, where driving scenarios (e.g., highway merging) involve asymmetric traffic patterns. Brightness variation, which mimics lighting changes, significantly improved performance, particularly for PilotNet Branched, which achieved the lowest steering angle Huber loss of 0.00031. This enhancement is attributable to the technique's ability to prepare models for pixel intensity shifts in varied illumination conditions, prevalent in InterchangeSE's dynamic scenarios (e.g., early morning, midday, or night driving). PilotNet and JNet also benefited, with Huber losses of 0.00045 and 0.00055, respectively, indicating robustness to lighting variations critical for real-world AV deployment. Simulating sensor imperfections, the third data augmentation technique (i.e., Gaussian noise) improved robustness, with PilotNet Branched achieving a steering angle Huber loss of 0.00033. PilotNet followed closely at 0.00052, while JNet's performance of 0.00057 was less pronounced, likely due to its simplified feature extraction struggling with noisy inputs. The improvement suggests that noise augmentation enhances model generalization to degraded image quality, relevant for challenging scenarios like highway merging. Shadow manipulation, the fourth data augmentation technique responsible for replicating occlusions, yielded mixed results. PilotNet Branched performed best with a result of 0.00038, followed by PilotNet and then JNet. The moderate improvement indicates that shadow augmentation aids adaptation to obscured road features, such as lane markings in intersections, but its impact is less significant than brightness or noise augmentation, possibly due to the controlled lighting in InterchangeSE.

The models were deployed and evaluated in three main driving scenarios to evaluate their effectiveness: 1) three-lane highways, 2) highway merging, and 3) to evaluate their effectiveness. PilotNet Branched excelled in three-lane highways, leveraging its multi-task learning to maintain low steering errors in stable conditions. In highway merging, PilotNet Branched models, training using brightness data augmentation, demonstrated robust performance by handling dynamic traffic patterns effectively. However, in residential sectors, all models exhibited higher errors, with JNet struggling most due to its limited capacity to process complex scenes with frequent turns and stop-and-go traffic.

Table 2. Effects of data augmentation techniques applied to model architectures during training

Model Architecture	Augmentation Type	Steering Angle Huber Loss	Throttle Response Huber Loss
PilotNet	Flip	0.00114	0.0167
PilotNet	Brightness	0.00045	0.0167
PilotNet	Noise	0.00052	0.0158
PilotNet	Shadow	0.00057	0.0184
PilotNet Branched	Flip	0.00113	0.0179
PilotNet Branched	Brightness	0.00031	0.0151
PilotNet Branched	Noise	0.00033	0.0173
PilotNet Branched	Shadow	0.00038	0.0188
JNet	Flip	0.00137	0.0209
JNet	Brightness	0.00055	0.0194
JNet	Noise	0.00057	0.0190
JNet	Shadow	0.00065	0.0228

To address the efficacy of training models without traffic, a subset of models was trained on a traffic-free dataset and evaluated in a traffic-inclusive environment. PilotNet Branched models trained without traffic achieved a steering angle Huber loss of approximately 0.0005 in traffic-free tests but degraded to 0.0012 in traffic-inclusive scenarios. This performance drop underscores the importance of training with traffic to capture realistic interactions, as traffic-free data lacks the dynamic variability (e.g., lane changes, vehicle proximity) present in InterchangeSE's traffic-inclusive scenarios.

CONCLUSIONS AND FUTURE WORK

The evaluation results of the three ML architectures (i.e., PilotNet, PilotNet Branched, and JNet) within the InterchangeSE XR simulation platform demonstrates the potential of extended reality (XR) environments for training ML models for autonomous vehicle (AV) control by replicating human driving behavior. It is important to note that data was collected, and ML models trained and evaluated, all within a virtual environment. No real-world vehicles needed to be used at any point making these experiments much safer, especially for data collection. In addition, the use of an XR environment (i.e., InterchangeSE) allowed any scenarios to be replicated and captured whereas in the real-world this is not always safe or feasible.

Specifically, PilotNet Branched exhibited superior performance, particularly when utilizing brightness and noise augmentation during training, highlighting the efficacy of multi-task architectures in handling diverse driving scenarios. The significant improvement from brightness and noise augmentation underscores the importance of simulating varied lighting conditions, which is critical for applications containing unpredictable environments and perceptual sensor imperfections. A notable limitation of this study is the absence of a standardized dataset to establish a ground truth comparison for the ML architectures. The custom dataset, comprising 303,980 samples collected from InterchangeSE, while representative of the simulated scenarios, lacks the standardized benchmarks that enable direct comparison with other studies. This restricts the ability to contextualize the performance of PilotNet, PilotNet Branched, and JNet against established baselines in the field, potentially limiting the generalizability of the findings. In addition, the performance degradation observed in traffic-inclusive scenarios for models trained on traffic-free data underscores the necessity of training datasets that reflect the complexity of operational environments to ensure robustness. This is particularly relevant, where simulations must incorporate realistic threats, such as adversarial vehicles, to prepare models for high-risk logistics scenarios. JNet's challenges in residential sectors further indicate that lightweight architectures may require additional tuning or hybrid approaches to effectively manage complex, low-speed scenarios. Future work will address these limitations by incorporating standardized datasets to provide a ground truth comparison for the evaluated ML architectures. This will enable benchmarking against established baselines, enhancing the validity and generalizability of the results. In addition, efforts will focus on expanding the diversity of the training dataset to include a broader range of maneuvers and environmental conditions, mitigating biases observed in the current dataset.

REFERENCES

Alvarez, S. (2024). Tesla investment in self-driving program to exceed $10B this year: Musk. *Teslarati*. https://www.teslarati.com/tesla-self-driving-program-investment-over-10b-2024-musk

AUVSI. (2024). *FY 2024 DoD Budget Report*. Association for Unmanned Vehicle Systems International.

Behringer, R., Sundareswaran, S., Gregory, B., Elsley, R., Addison, B., Guthmiller, W., Daily, R., & Bevly, D. (2004). The DARPA grand challenge-development of an autonomous vehicle. *IEEE Intelligent Vehicles Symposium, 2004*, 226–231.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., & Zieba, K. (2016). *End to End Learning for Self-Driving Cars* (arXiv:1604.07316). arXiv. http://arxiv.org/abs/1604.07316

Buehler, M., Iagnemma, K., & Singh, S. (2007). *The 2005 DARPA grand challenge: The great robot race* (Vol. 36). Springer Science & Business Media.

Cordesman, A. H., & Lin, A. (2015). *The IED Threat: Lessons from Iraq and Afghanistan*. Center for Strategic and International Studies. https://www.csis.org/analysis/ied-threat-lessons-iraq-and-afghanistan

Cubuk, E. D., Zoph, B., Mané, D., Vasudevan, V., & Le, Q. V. (2019). AutoAugment: Learning Augmentation Strategies From Data. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 113–123. https://doi.org/10.1109/CVPR.2019.00020

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. (2017). CARLA: An Open Urban Driving Simulator. In S. Levine, V. Vanhoucke, & K. Goldberg (Eds.), *Proceedings of the 1st Annual Conference on Robot Learning* (Vol. 78, pp. 1–16). PMLR. https://proceedings.mlr.press/v78/dosovitskiy17a.html

Espié, E., Guionneau, C., Wymann, B., Dimitrakakis, C., Coulom, R., & Sumner, A. (2005). *TORCS, The Open Racing Car Simulator*. https://api.semanticscholar.org/CorpusID:16920486

Fellendorf, M., & Vortisch, P. (2010). *Microscopic Traffic Flow Simulator VISSIM*. https://api.semanticscholar.org/CorpusID:59878753

Gray, M. (2025). Follow the Money: What the Pentagon's Budget Says About Preparing for the Future of Warfare. *Gray Matters*. https://maggiegray.us/p/follow-the-money-what-the-pentagons

Kocić, J., Jovičić, N., & Drndarević, V. (2019). An end-to-end deep neural network for autonomous driving designed for embedded automotive platforms. *Sensors*, *19*(9), 2064.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., & Wiessner, E. (2018). Microscopic Traffic Simulation using SUMO. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2575–2582. https://doi.org/10.1109/ITSC.2018.8569938

Mikołajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, 117–122. https://doi.org/10.1109/IIPHDW.2018.8388338

National Highway Traffic Safety Administration. (2022). *Automated Vehicle Testing: Safety and Performance Standards* [Report]. U.S. Department of Transportation. https://www.nhtsa.gov/vehicle-manufacturers/automated-driving-systems

NVIDIA. (2024). *NVIDIA DRIVE Sim: A Simulation Platform for Autonomous Vehicles*. https://developer.nvidia.com/drive/simulation

O'Kane, S. (2024). Waymo's latest funding round boosts it to a $45B valuation. *TechCrunch*. https://techcrunch.com/2024/11/05/waymos-latest-funding-round-boosts-it-to-a-45b-valuation/

Pomerleau, D. A. (1988). ALVINN: An Autonomous Land Vehicle in a Neural Network. In D. S. Touretzky (Ed.), *Advances in Neural Information Processing Systems 1 (NIPS 1988)* (pp. 305–313). Morgan Kaufmann. https://papers.nips.cc/paper/1988/file/812b4ba287f5ee0bc9d43bbf5bbe87fb-Paper.pdf

Precedence Research. (2025). *Autonomous Vehicle Market Size to Worth USD 4450.34 Billion by 2034*. Precedence Research. https://www.precedenceresearch.com/autonomous-vehicle-market

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(1), 1–48.

Unity Technologies. (2005). *Unity Game Engine*. https://unity.com/

U.S. Army. (2021). *Robotic and Autonomous Systems Strategy*. U.S. Army Training and Doctrine Command. https://www.tradoc.army.mil/wp-content/uploads/2021/03/RAS_Strategy.pdf

U.S. Department of Defense. (2023). *Fiscal Year 2024 Budget Request for Artificial Intelligence*. DoD Office of the Chief Information Officer. https://www.defense.gov/News/Releases/Release/Article/3346145/dod-announces-fiscal-year-2024-budget-request/

World Health Organization. (2023). *Road traffic injuries: Key facts*. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries