An Initial Exploration of Employing Large Multimodal Models in Defending Against Autonomous Vehicles Attacks

Mohammed Aldeen*, Pedram MohajerAnsari*, Jin Ma, Mashrur Chowdhury, Long Cheng, Mert D. Pesé
School of Computing, Clemson University
{mshujaa, pmohaje, jin7, mac, lcheng2, mpese}@clemson.edu

***Abstract*— As the advent of autonomous vehicle (AV) technology revolutionizes transportation, it simultaneously introduces new vulnerabilities to cyber-attacks, posing significant challenges to vehicle safety and security. The complexity of these systems, coupled with their increasing reliance on advanced computer vision and machine learning algorithms, makes them susceptible to sophisticated AV attacks. This paper explores the potential of Large Multimodal Models (LMMs) in identifying Natural Denoising Diffusion (NDD) attacks on traffic signs. Our comparative analysis show the superior performance of LMMs in detecting NDD samples with an average accuracy of 82.52% across the selected models compared to 37.75% for state-of-the-art deep learning models. We further discuss the integration of LMMs within the resource-constrained computational environments to mimic typical autonomous vehicles and assess their practicality through latency benchmarks. Results show substantial superiority of GPT models in achieving lower latency, down to 4.5 seconds per image for both computation time and network latency (RTT), suggesting a viable path towards real-world deployability. Lastly, we extend our analysis to LMMs' applicability against a wider spectrum of AV attacks, particularly focusing on the Automated Lane Centering systems, emphasizing the potential of LMMs to enhance vehicular cybersecurity.**

I. Introduction

In the past few years, autonomous vehicle (AV) systems witnessed great success of deep neural networks (DNNs) in a variety of computer vision tasks, such as image classification, object detection, etc. These advanced models have become increasingly robust against a multitude of AV attacks. For instance, techniques that use shadows [1] or stickers [2] to deceive traffic sign detection systems in autonomous vehicles have been effectively countered by the enhanced capabilities of DNNs [3]. The evolution of DNNs has improved traffic sign recognition accuracy, significantly boosting autonomous vehicles' safety and efficiency.

However, with the advancement of diffusion models in image generation, they reveal new vulnerabilities that could be a challenge for the robust detection capabilities of the existing DNN models [4]. For example, innovations, such as OpenAI's DALL-E [5], Adobe Firefly [6], and the VQ-GAN + CLIP [7] combination have redefined image generation, seamlessly converting text descriptions into detailed, photorealistic images. Images from these models pose a threat to AV systems, especially with Natural Denoising Diffusion (NDD) attacks [8], a new cybersecurity challenge for AVs.

*The first two authors contributed equally and are ordered alphabetically.

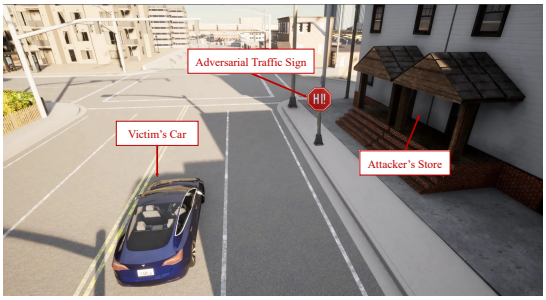

Fig. 1. The victim's car is approaching a pole in front of the attacker's store with an adversarial traffic sign depicting the word *'HI!'*. In our experiments, this sign has been classified as a *stop sign* by ResNeXt with a confidence score of 95%.

Attackers can use diffusion models to create images that, while not actual traffic signs, deceive AV perception systems' DNNs into recognizing them as real traffic signs. Since it is illegal to use, alter, or replicate official traffic signs [9], attackers can use NDD attacks to manipulate AV behavior without legal risks associated with physically tampering or using authentic road signs. An attacker could generate a fake *Stop* sign, visually distinct yet recognized by an AV's perception system, avoiding law enforcement attention but potentially causing the AV to stop unexpectedly, leading to confusion or accidents.

The advent of Large Language Models (LLMs) marks a significant milestone in Artificial Intelligence and Large Multimodal Models (LMMs) [10], [11] expand the capabilities of LLMs by incorporating visual signals. LMMs excel not only in handling and generating substantial textual-only tasks, but also demonstrate impressive performance in various multimodal tasks, such as video recommendations and image understanding [12], amongst others. This paper presents an in-depth analysis of the robustness of LMMs against NDD attacks and their integration into AV systems. Given the success of generative pretraining in vision-language modeling, we use combined visual and textual data such as multimodal GPT-4V [13] and LLaVA [14]. This paper further discusses how the use of instruction tuning in LMMs was instrumental in identifying NDD attack sample images. In summary, we make the following contributions:

- We conduct comprehensive evaluation of large multimodal models (LMMs) in identifying traffic signs compromised by NDD attacks. Our findings show that LMMs, such as GPT-4V and Google Bard showed accuracies of 84.06% and 85.42% respectively, outper-

forming state-of-the-arts models such as ResNeXt and MobileNet, which had much lower accuracies (below 18.67%). [1]
- We integrate LMMs into constrained computing environments, which are common in AVs, to demonstrate their potential feasibility in real-world scenarios. Our results illustrate that the latency was significantly high when these models were run locally. Conversely, server-run models like GPT-4V reduced latency to about 4.5 seconds, enhancing their practicality in AV systems.
- We expand the usage of LMMs' defensive capabilities against other AV attacks beyond traffic sign detection. We evaluate the effectiveness of LMMs in detecting road markings in the presence of adversarial patches in Automated Lane Centering (ALC) systems.

II. Related Work

Mao *et al.* [15] introduced a novel approach to autonomous driving by integrating OpenAI's GPT-3.5 model into a vehicle's motion planner. By treating motion planning as a language modeling problem and using language tokens for input and output, the study effectively transforms the GPT model into a planner that can also explain its decisions in natural language. Successful experiments on the nuScenes dataset showcase the approach's effectiveness and interpretability, highlighting its potential to enhance autonomous driving with advanced language model capabilities. Yang *et al.* [16] discussed using LLMs to enhance human-centric autonomous systems for interpreting user commands, focusing on complex and emergency scenarios in autonomous vehicles. It investigates various LLMs' efficacy and prompt designs in a few-shot multivariate binary classification. The findings confirm LLMs' general capability to comprehend and logically process prompts.

Chen *et al.* [17] show that LMMs are utilized in autonomous driving for enhancing context understanding and decision-making through a novel object-level multimodal architecture that merges vectorized modalities with pre-trained language models. Zarzà *et al.* [18] show improved traffic accident prediction using deep learning and introduce real-time interventions with compact large language models, enhancing autonomous driving systems for safer smart city planning. Our exploration of LMMs in the realm of AVs marks a significant advancement in automotive technology.

A. Threat Model

An attacker can get the same Traffic Sign Recognition (TSR) module as in the victim's vehicle to comprehend its implementation fully. This can involve buying or leasing the same car model as the victim's and then reverse engineering it, a method proven feasible with Tesla's Autopilot [19]. Additionally, it's worth noting that some TSR module algorithms used in production are open-source [20]. Using the white-box knowledge, the attacker creates and places an adversarial traffic sign with the text *'HI!'* on a pole across from their store, as illustrated in Figure 1. The victim's vehicle which is headed towards the store will recognize the fake traffic sign as a *stop sign* and come to a halt.

The goal of the adversary is to minimize the *obviousness* of the generated traffic sign to increase stealthiness. To achieve this, context-aware adversarial example generation is recommended. For instance, the adversary might want to install an adversarial traffic sign resembling the image of a vegetable near a grocery store. We assume the attacker exclusively targets the TSR module using AI-generated signs, without considering patches or alternative threats.

III. Datasets and Models

A. Dataset

To systematically assess the effectiveness of LMMs in identifying these NDD samples as adversarial, we first use the Adobe Firefly diffusion model [6] to generate a small-scale dataset containing NDD adversarial examples. We used text prompts aimed to disrupt the fundamental properties that humans typically use to identify these signs. For example, we focused on altering the most important visuals of traffic signs, changing their shape, texture, and color. These elements are crucial for how objects are typically recognized, as emphasized in existing research [21]. To generate a diverse set of samples, we created combinations such as altering both shape and text, shape and pattern, alongside other combinations, as depicted in Table I. Subsequently, two of the authors filtered the dataset manually to ensure that the generated images do not reflect actual traffic signs.

The generated dataset features images of four common traffic signs from the German Traffic Sign Recognition Benchmark (GTSRB) [22], namely *no entry* 2a, *priority road* 2b, *stop sign* 2c, and *yield* 2d, 40 variations were generated from each of the 4 real signs, resulting in a total of 160 signs. Each sign was changed in 4 features, resulting in 10 variations for each feature. We validated the adversarial effectiveness of our NDD dataset by testing with the ResNeXt model, including only images predicted as traffic signs with over 80% confidence, to assess the risk of NDD attacks misleading autonomous driving systems.

(a) No Entry (b) Priority Sign (c) Stop Sign (d) Yield Sign

Fig. 2. The Selected Traffic Signs

B. Models

To thoroughly evaluate the generated NDD dataset, we employed a different set of models. Our selection includes state-of-the-art models from conventional deep neural networks paradigms, such as pre-trained ResNeXt model that had been trained on GTSRB dataset [23], and manually trained MobileNet [24], VGG16 [25], YOLOv5 [26], serving as a baseline for our comparisons. On the other hand,

[1]The implementations codes of this work and generated dataset are available at `https://github.com/moaldeen/LMM_on_AV`.

TABLE I

A DETAILED BREAKDOWN OF HOW WE GENERATE THE NDD DATASET. WE ELIMINATE OR MODIFY SOME OR ALL OF THE FOUR FEATURES. THE RESNEXT MODEL IS USED TO CALCULATE THE PREDICTED CLASS AND CONFIDENCE SCORE.

Features	Generated Sample	Predicted Class + Confidence	Combination Features	Generated Sample	Predicted Class + Confidence
Shape		Prediction: No Entry Confidence: 99.7%	Shape & Text		Prediction: Stop Sign Confidence: 98.9%
Color		Prediction: No Entry Confidence: 99.3%	Shape & Pattern		Prediction: Stop Sign Confidence: 99.9%
Text		Prediction: Stop Sign Confidence: 99.9%	Color & Pattern		Prediction: No Entry Confidence: 93.44%
Pattern		Prediction: Yield Confidence: 95.5%	Shape & Text & Color & Pattern		Prediction: Yield Confidence: 85.07%

TABLE II

TEXT PROMPT FORMAT AND EXAMPLES FOR GENERATING AES FROM "STOP SIGN".

	Prompt for Generating AEs	
Altered Feature	**Prompt Format**	**Example: Stop sign**
-	Generate [Traffic Sign]	Stop sign
Shape	Generate [Shape] [Traffic Sign]	Circular Stop sign
Color	Generate [Color] [Traffic Sign]	Black Stop sign
Text	Generate [Traffic Sign] with a [Text] on it	Stop sign with "Hi" on it
Pattern	Generate [Traffic Sign] with a [Pattern] on it	Stop sign with a dotted pattern on it

we incorporated Large Multimodal Models (LMMs) such as GPT-4V [13], LLaVA-7B, LLaVA-13B [14], as well as Google Bard [27]. For the testing process, all LMMs were employed in their pre-trained state without any further fine-tuning to assess their out-of-the-box efficacy against the NDD dataset.

LMMs are intensive and memory-demanding, posing challenges in environments with limited hardware, such as autonomous vehicles. Quantization [28] emerges as a vital technique to reduce the precision of weight values, thereby conserving memory and accelerating the inference process, all with minimal impact on the performance of the model [29], as will be discussed in Section V. Therefore, we first convert the LLaVA models to a *fp16* binary format. The focal point of our quantization process is reducing the number of bits used to represent each weight in the model to *4-bits*.

IV. TASK 1: NDD ATTACK DISCOVERY

A. Goal

In this task, LMMs are asked to discover if images in the generated NDD dataset are related to actual traffic signs to evaluate their practical applicability in real-world scenarios. AVs, for example, use traffic sign recognition models as part of their navigation systems. Through this comprehensive evaluation, we aim to understand how different models react to the NDD attack and assess the robustness of the LMMs model against NDD attacks.

B. Experimental setup

We formulate NDD attack discovery as a binary-class classification task. Given an NDD sample image from the dataset, we ask the LMM model via prompt whether the image corresponds to actual traffic sign. Here is the prompt used in this scenario: *"**Q1**: Is the traffic sign displayed a real-world traffic sign that has the same shape, color, pattern and text as real world traffic sign ? Answer with 'yes' or 'no'."*. Then, we enumerate the AE images in the dataset, systematically presenting each to the LMMs for classification.

Similarly, we evaluate traditional traffic sign detection models, ResNeXt, MobileNet, YOLOv5, VGG16 in identifying traffic signs within the NDD dataset. To thoroughly assess each model's performance, we not only obtained the inference results from each image in the dataset but also focused on acquiring the confidence scores of each classified image since a high confidence score in classifying a NDD sample can reveal the model's susceptibility to such attacks. For instance, if a model wrongfully classifies a non-actual traffic sign (from our generated NDD dataset) as a legitimate traffic sign with high confidence, it indicates a potential vulnerability in the model's detection capabilities.

C. Results

Table III presents the overall detection results of all four signs in the generated NDD dataset. Rather than measuring the misclassified traffic sign class, the table evaluates if each model identifies NDD signs as a legitimate traffic sign or not. For example, the accuracy in Table III reflects the model's

TABLE III
COMPARISON OF LLMS AND TRADITIONAL MODELS AGAINST NDD ATTACK SAMPLES

Type	Model	Accuracy	F1-Score	Precision	Recall
LLMs	GPT-4V	84.06%	**86.73%**	**92.42%**	84.06%
	Bard	**85.42%**	85.94%	86.82%	**85.42%**
	LLaVA-7B	79.80%	79.93%	80.88%	79.80%
	LLaVA-13B	80.81%	83.48%	88.70%	80.81%
Traditional Models	ResNeXt [23]	17.17%	6.34%	3.99%	17.17%
	MobileNet [24]	18.67%	22.41%	62.50%	15.01%
	YOLOv5 [26]	44.12%	43.57%	65.11%	38.22%
	VGG16 [25]	**71.05%**	**66.56%**	**71.10%**	**78.88%**

capability to identify samples in the NDD dataset as non-actual traffic signs. We observe GPT and Bard to exhibit the highest accuracy achieving 84.06% and 85.42%, respectively. LLaVA-7B and LLaVa-13B also demonstrate noteworthy performance with accuracies of 79.80% and 80.81%, respectively. While the LLaVA models are effective in identifying NDD samples as non-actual traffic signs, they are slightly outperformed by GPT and Bard. On the other hand, traditional DNN models such as ResNeXt and MobileNet and YOLOv5 show significantly lower accuracy in identifying NDD samples, with accuracies of 17.17%, 18.85% and 44.12%, respectively. Nonetheless, VGG16 emerges as an exception among traditional models, achieving a noteworthy accuracy of 71.05% and the highest F1-Score in its category, indicating a relatively better but still not comparable performance to LMMs.

This notable success in LLMs is largely due to the their ability to handle complex visual patterns due to their extensive training on diverse datasets. LMMs have an advanced understanding of context. This means they are better at interpreting the broader meaning or implications of the data they process, rather than just focusing on specific features. This capability makes them more effective at identifying anomalies or irregularities in data, which is crucial for detecting and responding to attacks, where data might be intentionally altered to mislead the model. On the contrary, traditional DNN models rely heavily on visual cues or specific features in the data they are trained on such as shape, color, and text. These models have been optimized to identify these features with high accuracy under normal conditions. In the case of NDD attacks, these visual features are subtly manipulated so that traditional models still continue to predict the presence of traffic signs with high confidence. This overconfidence is likely due to the altered signs still retaining enough of the original features to trigger recognition by the model. The LLMs vary in false positive rates, with Bard at 8.33%, LLama-7b at 10.10%, GPT at 14.49%, LLama-13b at 16.16%, and Resnex with the highest rate at 82.83%.

Despite the strong performance of Google Bard, integrating it was tough due to no official API. We used an unofficial API [30], which worked but had limits, especially handling lots of images. It couldn't process batches over 30 images well, even with delays. So, we only used this method for Task 1, leaving Bard out of Table IV and in Task 2.

TABLE IV
COMPARISON OF LMMS AND TRADITIONAL MODELS AGAINST NON-AES FOR GTSRB DATASET

Type	Model	Accuracy	F1-Score	Precision	Recall
LMMs	GPT-4V	**79.04%**	**84.79%**	**95.06%**	**79.04%**
	LLaVA-7B	73.00%	69.91%	80.92%	73.00%
	LLaVA-13B	70.00%	62.39%	59.66%	70.00%
Traditional Models	ResNeXt [23]	**99.50%**	**99.50%**	**99.51%**	**99.50%**
	MobileNet [24]	93.50%	96.44%	99.99%	93.50%
	YOLOv5 [26]	54.80%	56.86%	94.67%	52.81%
	VGG16 [25]	99.00%	99.20%	99.22%	99.20%

V. TASK 2: LMM INTEGRATION IN AV PERCEPTION

A. Goal

Building upon the insights gained from Task 1, where LMMs demonstrated a notable proficiency in identifying the images within our generated NDD dataset as non-actual traffic signs, the goal of this task is to explore the feasibility of integrating LMMs into the perception systems of autonomous vehicles to enhance decision-making and environmental understanding. We examine how the integration of LMMs, with their significant computational requirements, aligns with the operational capabilities of AVs, aiming to strike an optimal balance between enhanced cognitive processing and the computational efficiency of onboard vehicle systems.

B. Method

In the **first** phase, we integrated ZED BOX [31], designed for running sophisticated neural networks and processing voluminous 3D sensor data in real-time, which is crucial for the complex decision-making processes of autonomous vehicles. It allows to run modern neural networks and process 3D sensor data in real-time . The ZED BOX is equipped with the latest JetPack, along with CUDA, TensorRT, and CuDNN libraries, making it a robust platform for advanced computing tasks. Furthermore, one of the major tasks in perception systems is object detection, which is essential for safe navigation without collisions. The ability to detect objects and take appropriate actions, such as braking or path adjustment, relies heavily on assessing the distance to potential obstacles. To achieve this, we employ a stereo camera system, which, by featuring two camera sensors, mimics human binocular vision and can capture three-dimensional images. Specifically, we utilized the ZED X – an IP66-rated stereo camera powered by the Neural Depth Engine 2, designed for next-generation robotics and ideally suited for industrial environments. This camera employs triangulation to construct a three-dimensional understanding of the scene, thereby significantly improving our perception of space and motion within the test environment.

In the **second** phase, we utilize the Raspberry Pi as an Electronic Control Unit (ECU) to simulate an autonomous vehicle's perception system, particularly focusing on its response to NDD dataset. We chose the Raspberry Pi 4 Model B, which features 8GB RAM and 64GB ROM, a Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit SoC running at 1.8GHz, due to its similar specifications

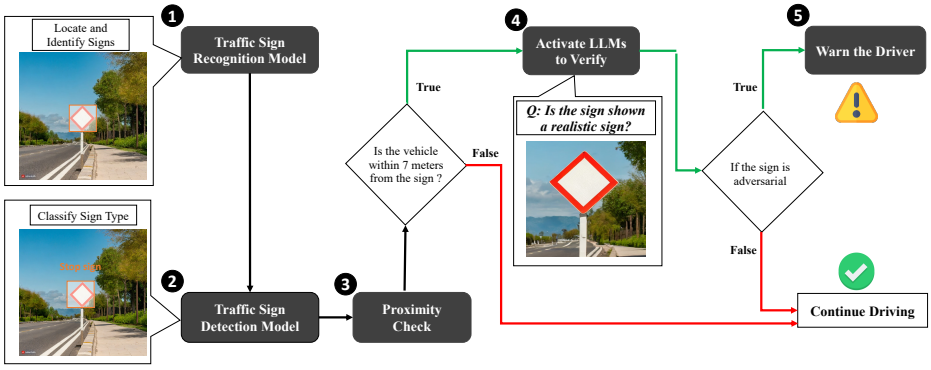

Fig. 3. Workflow of the LLM as a Verification Layer in an Autonomous Vehicle's Perception System.

to comparable autonomous driving ECUs. For instance, while not as advanced as the high-computing Tesla HW3 or HW4, the Raspberry Pi 4's processing capabilities and system architecture offer a modest parallel to early versions of Tesla's Autopilot hardware, such as HW1. Specifically, Raspberry Pi 4 provide a sufficient platform for handling tasks such as image recognition and processing sensor data, similar to the capabilities of the Mobileye EyeQ3 chip in Tesla's HW1 Moreover, it allows us to create a controlled environment where LMMs can be tested on their ability to process environmental data, including images from the NDD dataset.

Traditional traffic sign detection models based on DNNs have proven to be effective in accurately recognizing non-adversarial traffic signs, as shown in Table IV. Meanwhile, LMMs demonstrate promising capabilities in identifying NDD attack examples. Therefore, we deploy LMMs in our system as a verification layer, and the workflow of this approach is illustrated in Figure 3. This integration forms a comprehensive perception system for street view analysis in AVs. Initially, a traffic sign recognition model scans the street to locate and identify traffic signs then models such as ResNeXt are utilized to classify the specific type of traffic sign detected. When the vehicle approaches within a proximity of 7 meters to the traffic sign, LMMs are activated as a verification layer to confirm whether the identified sign is non-adversarial. Based on the outcome of the LMM verification, the system either warns the driver if the sign is deemed adversarial or allows the vehicle to continue driving if the sign is verified as legitimate.

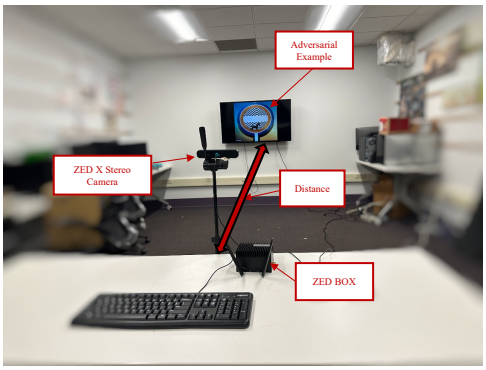

Fig. 4. Experimental Setup

To determine this operational range of 7 meters for LMM activation, we conducted a series of tests at varying distances. These tests were designed to identify the threshold distance at which the LMMs reliably and accurately identified traffic signs as adversarial or legitimate. We initiated our testing process by capturing images (via ZED stereo camera) of traffic signs from a close distance of 2 meters. LMMs were capable of accurately identifying NDD signs in a short-range scenario. Following that we tested at 5, 7, and finally 10 meters. The accuracy of the LMMs in identifying NDD signs was high up until 7 meters. However, beyond this threshold, it dropped significantly, leading us to set 7 meters as the optimal activation range for the LMMs verification layer. The observed decrease in accuracy beyond the threshold can be related to the limited capability of the camera to capture clear and detailed images at longer distances, leading to a reduction in the quality of data fed to the LMMs. Additionally, the nature of the signs being displayed on a TV screen may have influenced the captured image quality. This finding highlights the importance of considering both the technological capabilities and the operational environment in determining the effective range for LMM activation, ensuring optimal threat detection and adequate response time for the AV's decision-making processes.

C. Experimental setup

We connect the ZED BOX to the ZED X stereo camera module through its GMSL2 port to capture real-time images from the NDD dataset projected on a TV, providing a dynamic and realistic testing environment as shown in Figure 4. We deployed GPT-4V on it for testing the NDD dataset. The captured images from the camera module are directly fed into the GPT-4V model as we formulate NDD attack identification with a binary classification prompt. The prompt used in this scenario is: "***Q2**: Is the traffic sign shown a real-world sign commonly used in the physical environment, such as on roads, highways, or streets? If 'yes', name the sign in three words. If 'no', simply respond with 'no'*".

However when working with LLaVA models, we first need to quantize the models as outlined in Section III-B. This quantization is crucial for running these models on the hardware of a ZED BOX. Following this, we apply the same method used for the GPT-4V model by feeding images captured by ZED X stereo camera into the quantized LLaVA

TABLE V
COMPARISON OF RPI AND ZED BOX AGAINST AES

Device	Model	Accuracy	F1-Score	Precision	Recall
ZED BOX	GPT-4V	77.27%	48.28%	35.00%	77.78%
	LLaVA-7B	59.09%	18.18%	12.50%	33.33%
	LLaVA-13B	53.03%	34.04%	21.05%	88.89%
Raspberry Pi	GPT-4V	74.24%	45.16%	31.82%	77.78%
	LLaVA-7B	59.10%	30.08%	20.00%	66.70%
	LLaVA-13B	48.48%	29.16%	17.94%	77.77%

models and we employed the same binary classification prompt for NDD attack identification. This direct approach enables us to evaluate the model's performance in real-time, mirroring potential real-world applications in AVs.

In a parallel experiment, we replicated this setup using a Raspberry Pi 4 Model B. This experiment aimed to assess the portability and efficiency of our approach on more constrained hardware environments. It is worth noting that the Raspberry does not have the GMSL2 port to connect the stereo camera. Therefore, we utilized images that were previously captured using the ZED BOX and then fed directly into the Raspberry. This approach ensured consistency in the testing environment across different hardware platforms.

D. Results

Table V illustrates the comparative performance of the Raspberry Pi and ZED BOX in handling LMMs to detect NDD attack samples. Notably, the GPT models on the ZED BOX and Raspberry Pi demonstrate higher accuracy of 77.27% and 74.24%, respectively. This observation aligns with the outcomes presented in Task 1, as illustrated in Table III, where GPT models emerge as the superior among other LMMs. Despite using the same GPT environment and images as deployed on the ZED BOX, the GPT model on Raspberry Pi shows a slight decline in its performance.

Meanwhile, the LLaVA models displayed varying performances, with LLaVA-7B surprisingly outperforming the more robust LLaVA-13B model. This unexpected outcome may stem from the inherent complexity and quantization tolerance of each model. LLaVA-7B, being less complex than LLaVA-13B, might be more resilient to the precision loss from quantization, retaining more effectiveness. This means that when the models are converted to a lower precision format (such as *fp16* binary format) for deployment in environments with limited hardware and memory capacity such as AVs, the less complex LLaVA-7B model retains more of its effectiveness compared to its more complex variant.

Figure 5 analyzes the latency in processing LMMs to detect NDD attack samples. GPT-4V has the shortest processing time among the three models for both devices averaging around 4.5 seconds per image. One the other hand, LLaVA-13B shows a significant latency on the Raspberry, with processing times exceeding 30 minutes per image, while the ZED BOX processes the same model in roughly 10 minutes per image. LLaVA-7B demonstrates considerably lower latency on both devices, with the Raspberry taking around 10 and the ZED BOX about 5 minutes per image.

The substantial latency difference between the LLaVa and GPT models is associated with the computational complexity of LMM. The latency observed in GPT models is mainly network round-trip time (RTT) latency, where data is transmitted to and from the server for processing. In this scenario, the actual computation is being carried out on OpenAI servers. Unlike GPT, the LLaVA models require heavy computations to be performed directly on local hardware, leading to higher computational latency, especially in resource-constrained environments such as the Raspberry Pi or ZED BOX.

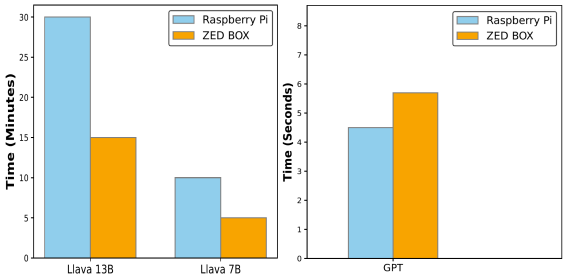

Fig. 5. Comparative Performance Analysis: Processing Times of ZED Box vs. Raspberry Pi

VI. TASK 3: GENERALIZING LMM DEFENSE AGAINST FURTHER ATTACKS

A. Goal

Building on Task 2's successful integration of LMMs into AV perception systems using Raspberry Pi and ZED BOX, this phase explores LMMs' potential to identify and mitigate attacks on Automated Lane Centering (ALC) systems. As shown in Figure 6, ALC adjusts steering to maintain lane centering but is vulnerable to attacks [32], [33], [34], compromising detection accuracy and leading to potential lane departures. This task assesses LMMs' effectiveness against such attacks targeting ALC systems.

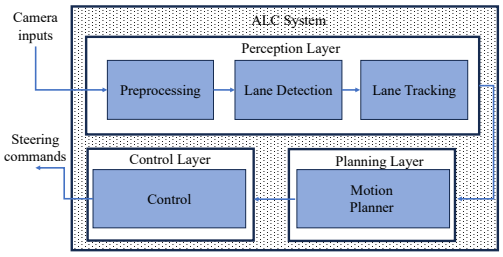

Fig. 6. Overview of a Common ALC System Design.

B. Method

To generate adversarial examples for attacking AV lane detection, we utilized the Dirty Road Patch (DRP) attack [32]. These patches disrupt the ALC system by mis-detecting lane markings when placed on roads. We inserted two 75-meter DRP patches into a CARLA simulator map [35]—an open-source tool for autonomous driving research, featuring realistic scenarios and sensor support. The efficacy of this attack was tested under ClearNoon and WetCloudSunset conditions. Adversarial patch visibility differed, as shown in Figure 7, with one lane faded and the other more distinct.

Fig. 7. Comparison of lanes (Faded Lane - Clear Lane)

TABLE VI
EFFECTIVENESS OF LLMS IN LANE MARKING DETECTION UNDER ADVERSARIAL PATCHES.

Model	Device	Lane Type	ClearNoon	WetCloudSunset
GPT-4V	ZED BOX	Faded Lane	100%	100%
		Clear Lane	100%	100%
	Raspberry Pi	Faded Lane	100%	100%
		Clear Lane	100%	100%
LLaVA-13B	ZED BOX	Faded Lane	44%	33%
		Clear Lane	60%	55%
	Raspberry Pi	Faded Lane	44%	33%
		Clear Lane	60%	55%
LLaVA-7B	ZED BOX	Faded Lane	33%	33%
		Clear Lane	50%	44%
	Raspberry Pi	Faded Lane	33%	33%
		Clear Lane	50%	44%

C. Experimental Setup

We replicated Task 2's environment to test LMMs on adversarial patches. The vehicle in CARLA navigated through the town to the adversarial patch location. As the car moved over the patch, the ZED X stereo camera module, connected to the ZED BOX, captured **10** images. Initially, these images were fed directly into GPT-4V, LLaVA-7B, and LLaVA-13B on the ZED BOX, with the prompt: "***Q3****: Assuming that you are a driver and see this view, what is your prediction on the lane, and as a driver, will you go 1. 'straight', 2. 'left', or 3. 'right'?"*. This experiment was then repeated using the Raspberry Pi 4 Model B under two different weather conditions.

D. Results

As shown in [32], placing the adversarial patch on the road caused the AV to mis-detect the lane marking, leading to collisions. Although 100% successful when applied for over 36 ms, tests show GPT-4V detects road markings accurately with the patch, as depicted in Table VI. LLaVA models, though not as effective as GPT-4V, reduce DRP attack impacts. Correct detection by LMMs prevents incorrect maneuvers, enhancing safety.

VII. DISCUSSION

A. Implications

There are numerous papers showing different AV attacks targeting Object Tracking [36], Traffic Light Detection [37], Object Detection [36], Camera Localization [37], and LiDAR Perception [38]. These attacks can cause malfunctions in perception systems, leading to severe consequences. Adversarial training, adversarial detection, input reconstruction, input denoising, classifier robustification, network verification, and a combination of multiple models, such as defensive distillation and ensembling, are some popular defense methods against perception attacks [39]. While the performances of these methods in attack scenarios can increase the resilience of machine learning and deep learning models, they are often not comparable to non-attack scenarios and are still vulnerable to unknown attacks [40]. As shown in Section VI, although the LMM has not been trained over these adversarial patches and is thus an unknown attack, the LMM appears to still detect the original lane markings correctly in the presence of these patches. Therefore, we expect that the impact of our research can go well beyond just improving lane detection and help AVs' perception systems defend against different unknown attacks. Due to the relative novelty of LMMs, only few studies [17] have demonstrated the application of LMMs in AVs. However, none have specifically utilized LMMs as an in-vehicle tool for helping the perception system with recognizing adversarial attacks. By leveraging the strengths of LMMs in environmental perception and decision-making and combining them with AVs' perception system to identify and react to road markers, these vehicles are becoming more adept at navigating complex driving scenarios. The result is a more robust and intelligent system that can adapt to a variety of challenges on the road, potentially increasing the safety and efficiency of AVs.

B. Limitations

In our project, we faced two main limitations. First, it was challenging to simulate a real-world environment and accurately show the AV's performance integrated with the LMM. Deploying the LMMs directly within a real car to detect adversarial traffic signs, as tested in Section V, was not feasible due to the limited power and hardware capabilities typically found in current AVs. In this context, Tesla's HW4 system, an advanced update to the Autopilot ECU, is specialized for autonomous driving with features like 20 ARM cores, 2 GPUs, three neural network processors, and 16GB RAM, all optimized for this purpose. In contrast, the ZED Box, a powerful AI computer with spatial computing capabilities, comes with 16GB RAM and an AI performance of 100 TOPS, making it well-suited for versatile, high-performance AI tasks, including running LMMs. Thus, a simulated setup was employed using a ZED BOX equipped with a ZED X Stereo camera, positioned in front of a television. The ZED BOX functioned as a stand-in for an AV's vision and perception system, while the television simulated the vehicle's external environment. However, this setup could not perfectly mimic the complex dynamics of real-world scenarios. It provided a controlled and simplified version of real-life scenarios but lacked the unpredictability and complexity typically encountered in actual driving situations. Additionally, creating a realistic scenario for evaluating the LMMs' ability to detect original lane markings in the presence of adversarial patches posed a significant challenge, as detailed in Section VI. To accurately observe the LMMs' impact on steering commands, their output needed to be sent into the planning and motion planning modules of an AV's autonomous driving stack which introduced considerable complexity due to the intricate nature of AVs' system.

VIII. Conclusion

The potential of LMMs to enhance the perception systems of AVs in adversarial scenarios has been discovered in our work. To systematically assess the effectiveness of LMMs' detection capabilities against diffusion model attacks, we generated a small-scale NDD attack dataset. To ensure the adversarial effectiveness of our dataset, we processed the generated NDD dataset through the ResNeXt model, selecting images where the model predicted with more than 80% confidence that these images do not represent real-world traffic signs, underscoring their use strictly in adversarial testing environments. This highlights the potential for sophisticated NDD attacks to mislead autonomous driving systems. We hope that our study and dataset will inform the autonomous vehicle community about the potential of LMMs in detecting attacks, thus enhancing vehicle safety and security.

Acknowledgment

This work is partially supported by South Carolina Research Authority and by the National Center for Transportation Cybersecurity and Resiliency (TraCR) (a U.S. Department of Transportation National University Transportation Center) headquartered at Clemson University, Clemson, South Carolina, USA. Any opinions, findings, conclusions, and recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of TraCR, and the U.S. Government assumes no liability for the contents or use thereof.

References

[1] Y. Zhong, X. Liu, D. Zhai, J. Jiang, and X. Ji, "Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 345–15 354.

[2] W. Jia, Z. Lu, H. Zhang, Z. Liu, J. Wang, and G. Qu, "Fooling the eyes of autonomous vehicles: Robust physical adversarial examples against traffic sign recognition systems," *arXiv preprint arXiv:2201.06192*, 2022.

[3] E. Lella, N. Macchiarulo, A. Pazienza, D. Lofù, A. Abbatecola, and P. Noviello, "Improving the robustness of dnns-based network intrusion detection systems through adversarial training," in *2023 8th International Conference on Smart and Sustainable Technologies (SpliTech)*. IEEE, 2023, pp. 1–6.

[4] M. S. Graham, W. H. Pinaya, P.-D. Tudosiu, P. Nachev, S. Ourselin, and J. Cardoso, "Denoising diffusion models for out-of-distribution detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2947–2956.

[5] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8821–8831.

[6] D. C. Epstein, I. Jain, O. Wang, and R. Zhang, "Online detection of ai-generated images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 382–392.

[7] K. Crowson, S. Biderman, D. Kornis, D. Stander, E. Hallahan, L. Castricato, and E. Raff, "Vqgan-clip: Open domain image generation and editing with natural language guidance," in *European Conference on Computer Vision*. Springer, 2022, pp. 88–105.

[8] T. Sato, J. Yue, N. Chen, N. Wang, and Q. A. Chen, "Intriguing properties of diffusion models: A large-scale dataset for evaluating natural attack capability in text-to-image generative models," *arXiv preprint arXiv:2308.15692*, 2023.

[9] "Section 119-4 - altering or defacing of traffic signs," https://casetext.com/regulation/south-carolina-code-of-regulations/chapter-119-university-of-south-carolina/article-1-general-regulations/section-119-4-altering-or-defacing-of-traffic-signs, accessed: 27 December 2023.

[10] Y. Zeng, C. Jiang, J. Mao, J. Han, C. Ye, Q. Huang, D.-Y. Yeung, Z. Yang, X. Liang, and H. Xu, "Clip2: Contrastive language-image-point pretraining from real-world point cloud data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 244–15 253.

[11] C. Cui, Y. Ma, X. Cao, W. Ye, Y. Zhou, K. Liang, J. Chen, J. Lu, Z. Yang, K.-D. Liao *et al.*, "A survey on multimodal large language models for autonomous driving," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 958–979.

[12] J. An, Z. Yang, L. Li, J. Wang, K. Lin, Z. Liu, L. Wang, and J. Luo, "Openleaf: Open-domain interleaved image-text generation and evaluation," *arXiv preprint arXiv:2310.07749*, 2023.

[13] Z. Yang, L. Li, K. Lin, J. Wang, C.-C. Lin, Z. Liu, and L. Wang, "The dawn of lmms: Preliminary explorations with gpt-4v (ision)," *arXiv preprint arXiv:2309.17421*, vol. 9, p. 1, 2023.

[14] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *arXiv preprint arXiv:2304.08485*, 2023.

[15] J. Mao, Y. Qian, H. Zhao, and Y. Wang, "Gpt-driver: Learning to drive with gpt," *arXiv preprint arXiv:2310.01415*, 2023.

[16] Y. Yang, Q. Zhang, C. Li, D. S. Marta, N. Batool, and J. Folkesson, "Human-centric autonomous systems with llms for user command reasoning," *arXiv preprint arXiv:2311.08206*, 2023.

[17] L. Chen, O. Sinavski, J. Hünermann, A. Karnsund, A. J. Willmott, D. Birch, D. Maund, and J. Shotton, "Driving with llms: Fusing object-level vector modality for explainable autonomous driving," *arXiv preprint arXiv:2310.01957*, 2023.

[18] I. de Zarzà, J. de Curtò, G. Roig, and C. T. Calafate, "Llm multimodal traffic accident forecasting," *Sensors*, vol. 23, no. 22, p. 9225, 2023.

[19] T. K. S. Lab, "Experimental security research of tesla autopilot," 2023. [Online]. Available: https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Research_of_Tesla_Autopilot.pdf

[20] Comma.ai, "Openpilot by comma.ai," 2023. [Online]. Available: https://www.comma.ai/openpilot

[21] Y. Ge, Y. Xiao, Z. Xu, X. Wang, and L. Itti, "Contributions of shape, texture, and color in visual recognition," in *European Conference on Computer Vision*. Springer, 2022, pp. 369–386.

[22] S. B. Wali, M. A. Hannan, A. Hussain, S. A. Samad *et al.*, "An automatic traffic sign detection and recognition system based on colour segmentation, shape matching, and svm," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[23] T. Zhou, Y. Zhao, and J. Wu, "Resnext and res2net structures for speaker verification," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 301–307.

[24] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[26] G. Jocher, "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," https://github.com/ultralytics/yolov5, Oct. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4154370

[27] S. Siad, "The promise and perils of google's bard for scientific research," 2023.

[28] M. Nagel, R. A. Amjad, M. Van Baalen, C. Louizos, and T. Blankevoort, "Up or down? adaptive rounding for post-training quantization," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7197–7206.

[29] B. Deiseroth, M. Meuer, N. Gritsch, C. Eichenberg, P. Schramowski, M. Aßenmacher, and K. Kersting, "Divergent token metrics: Measuring degradation to prune away llm components–and optimize quantization," *arXiv preprint arXiv:2311.01544*, 2023.

[30] D. Park, "Bard api," 2023, accessed: 2023-12-20. [Online]. Available: https://github.com/dsdanielpark/Bard-API

[31] Stereolabs, "Zed box - embedded ai computer with nvidia® jetson," https://www.stereolabs.com/products/zed-box, 2023, accessed: 2023-04-10.

[32] T. Sato, J. Shen, N. Wang, Y. Jia, X. Lin, and Q. A. Chen, "Dirty road can attack: Security of deep learning based automated lane centering under {Physical-World} attack," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 3309–3326.

[33] P. Jing, Q. Tang, Y. Du, L. Xue, X. Luo, T. Wang, S. Nie, and S. Wu, "Too good to be safe: Tricking lane detection in autonomous driving

with crafted perturbations," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 3237–3254.

[34] M. Aldeen, P. MohajerAnsari, J. Ma, M. Chowdhury, L. Cheng, and M. D. Pesé, "Wip: A first look at employing large multimodal models against autonomous vehicle attacks," in *ISOC Symposium on Vehicle Security and Privacy (VehicleSec '24)*, 2024.

[35] CARLA, "Carla simulator," 2023. [Online]. Available: https://carla.org

[36] L. Huang, C. Gao, Y. Zhou, C. Xie, A. L. Yuille, C. Zou, and N. Liu, "Universal physical camouflage attacks on object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 720–729.

[37] W. Wang, Y. Yao, X. Liu, X. Li, P. Hao, and T. Zhu, "I can see the light: Attacks on autonomous vehicles using invisible lights," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 1930–1944.

[38] J. Sun, Y. Cao, Q. A. Chen, and Z. M. Mao, "Towards robust {LiDAR-based} perception in autonomous driving: General black-box adversarial sensor attack and countermeasures," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 877–894.

[39] X. Wang, J. Li, X. Kuang, Y.-a. Tan, and J. Li, "The security of machine learning in an adversarial setting: A survey," *Journal of Parallel and Distributed Computing*, vol. 130, pp. 12–23, 2019.

[40] Z. Khan, M. Chowdhury, and S. M. Khan, "A hybrid defense method against adversarial attacks on traffic sign classifiers in autonomous vehicles," *arXiv preprint arXiv:2205.01225*, 2022.