2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

ODD-diLLMma: Driving Automation System ODD Compliance Checking using LLMs

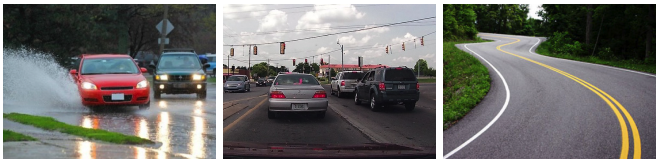
Carl Hildebrandt*[1], Trey Woodlief*[1], Sebastian Elbaum[1]

***Abstract*— Although Driving Automation Systems (DASs) are rapidly becoming more advanced and ubiquitous, they are still confined to specific Operational Design Domains (ODDs) over which the system must be trained and validated. Yet, each DAS has a bespoke and often informally defined ODD, which makes it intractable to manually judge whether a dataset satisfies a DAS's ODD. This results in inadequate data leaking into the training and testing processes, weakening them, and causes large amounts of collected data to go unused given the inability to check their ODD compliance. This presents a dilemma: How do we cost-effectively determine if existing sensor data complies with a DAS's ODD? To address this challenge, we start by reviewing the ODD specifications of 10 commercial DASs to understand current practices in ODD documentation. Next, we present ODD-diLLMma, an automated method that leverages Large Language Models (LLMs) to analyze existing datasets with respect to the natural language specifications of ODDs. Our evaluation of ODD-diLLMma examines its utility in analyzing inputs from 3 real-world datasets. Our empirical findings show that ODD-diLLMma significantly enhances the efficiency of detecting ODD compliance, showing improvements of up to 147% over a human baseline. Further, our analysis highlights the strengths and limitations of employing LLMs to support ODD-diLLMma, underscoring their potential to effectively address the challenges of ODD compliance detection.**

I. Introduction

Current *Driving Automation Systems* (DASs) [1] can only offer high-levels of autonomy under the limited operating conditions defined by their Operational Design Domain (ODD). SAE J3016 defines the ODD as "operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics" [1]. Yet, these ODDs are often incomplete and described imprecisely using natural language (NL). This limits our ability to align system design and validation techniques with a system's ODD.

In exploring the alignment between DASs' ODDs and real-world data, we confront a multifaceted challenge. ODDs encapsulate a range of conditions, as shown in Figure 1, that define when a DAS is designed to operate safely. However, the complexity and variability inherent in these domains pose significant hurdles for aligning ODDs and datasets. For example, an ODD may specify that operation is considered safe only in the absence of rain, where there are no traffic signals, or on non-winding roads. Each of these requirements, while straightforward to human comprehension, demands distinct checks when aligning with real-world data. Verifying the presence of rain for weather compatibility requires real-time environmental monitoring. Accurately identifying traffic signals requires a basic knowledge of road rules and an understanding of their relevance to various situations. Moreover, ascertaining whether a road does not have significant bends requires an analysis of the road's topology, potentially involving the classification of its layout or the calculation of specific metrics regarding its form. However, this detailed information is often not captured in existing datasets, leaving us the challenge of identifying these dimensions from the collected data, i.e. determining if the image depicts rain, traffic signals, or winding roads.



(a) Environment [5] (b) Traffic [6] (c) Roadway [7]

Fig. 1: Examples of potential ODD requirements.

Now consider that these complex requirements need to be applied to massive datasets of real-world sensor data used to train and test DASs. For example, in 2015 Tesla received 1 million miles of data every 10 hours [2]; in 2021, comma.ai stated that its users drive over 500,000 miles each week [3]; by 2023, Waymo had completed 7 million miles of fully autonomous driving [4]. Such rich field data cannot easily be applied at scale for DAS training or testing as the data may not comply with the system's ODD, raising the potential that ODD violations could leak into training or testing. The key questions we tackle are **1) how are ODDs specified today, and to what extent do existing datasets comply with their own ODD? 2) How can we automatically check existing datasets for compliance with respect to an ODD?**

To address the first question, we start by analyzing 10 commercial DASs to understand the current state of NL ODD specification. Then, we manually investigate 3 datasets of autonomous driving data, including 2 datasets utilized in training/testing a commercially-available, road-deployed DAS to judge their compliance with that DAS's ODD. Next, to address the second question, we explore an approach that integrates Large Language Models (LLMs) to automate the process of ODD compliance checking. LLMs have shown immense promise in their ability to mimic human compre-

*Equal contribution

[1]University of Virginia, United States, {hildebrandt.carl, adw8dm, selbaum}@virginia.edu

This work was funded in part by NSF through grants #2312487 and #2403060, and Lockheed Martin ATL. Trey Woodlief was supported by a University of Virginia SEAS Fellowship.

hension over NL and images [8]. Our proposed framework, ODD-diLLMma, leverages this ability, providing an automated and structured way to use LLMs to analyze DAS sensor datasets with respect to the NL specification of its ODD by producing for each input a "compliance vector" that characterizes the input's compliance with the ODD.

Our key contributions are: 1) An examination of commercial DAS ODDs, highlighting the challenges of consistent application and their misalignment with today's datasets. 2) The identification of LLMs as a foundational technology to address the disconnect between ODD specification and real-world data. 3) Definition of the ODD-diLLMma framework to leverage LLMs and ODDs specified in NL to enable analysis of real-world datasets. 4) A study implementing ODD-diLLMma with two state-of-the-art LLMs to analyze datasets for a commercially available DAS system, openpilot [9]. 5) An open-source[1] implementation of ODD-diLLMma.

II. Operational Design Domains

The ODD includes many factors, which we refer to as *"semantic dimensions"* of the ODD, e.g., raining, in intersection, etc. The full ODD is the product of all semantic dimensions. Prior ODD analysis for DAS has focused on improving ODDs [10], [11], deriving assurance cases [12], [13], and performing runtime restriction [14]. A few efforts aimed to utilize ODDs for DAS analysis and dataset generation [15], [16], [17]. However, *no techniques currently exist to automatically determine whether a dataset is compliant with an ODD*—we present the first steps to this end.

The next section provides a brief overview of the ODDs of 10 commercially available DASs to highlight the nuances in their specifications. Then, for one of those DAS, we illustrate the difficulties in checking dataset ODD compliance.

A. Examining ODDs on Deployed DASs

Kaiser et al. note that "while end users can understand an ODD definition best in a simplified natural language format, engineers need a more exact, in the best case formally underpinned definition of the ODD of the systems they are developing" [20]. However, such formalisms are not always available, with the end-user NL versions representing the *only publicly available* ODDs.

We inspected the NL ODD descriptions of the DASs offered by comma.ai [9], GMC [18], and Tesla [19] to extract selected semantic dimensions of the ODD shown in Table I. Tesla and comma.ai's DASs are composed of multiple components; each component has a separate ODD description. In the table, '×' indicates the semantic dimension is *excluded* from the ODD, while '✓' indicates *included* in the ODD; unless noted, no interpretation was done to identify the in/exclusion, i.e. the semantic dimension was stated directly.

From examining the DASs' ODDs we can draw several conclusions. The NL descriptions are often phrased to list what is excluded from the ODD rather than included, and each DAS has exclusions related to weather, e.g., GMC Super Cruise states that it should not be used "in adverse weather conditions, including rain", i.e. rain is out of ODD [18]. However, the NL descriptions used across systems are inconsistent, highlighting their imprecision. While GMC expressly excludes "rain", comma.ai more specifically excludes "heavy rain"; this makes ODD precise comparison and thus necessary train/test data difficult. GMC's system would require a dataset with no rain, and that same dataset could be used for comma.ai's systems since no rain implies no heavy rain; however, the reverse is not true. Further, since these ODD descriptions come from separate companies, comma.ai's definition of "heavy rain" may be equivalent to GMC's definition of "rain". The definitions' imprecision is further highlighted when examining the Tesla components where all but one expressly exclude "heavy rain" from their ODD; however, the Full Self-Driving (Beta) system excludes "rain" instead. Since these descriptions are all produced by Tesla, we may infer that "rain" for Full Self-Driving (Beta) is intended to have distinct semantics from "heavy rain" for the others. This suggests a continuum of "rain" intensity that is inadequately captured by the binary inclusion or exclusion within the ODD, reflecting the simplified manner in which complex conditions are represented.

This binary perspective, which categorizes conditions like rain as either "present" and *outside* the ODD or "absent" and *inside* the ODD, fails to capture the continuum of intensity inherent to such conditions and inadvertently introduces subjectivity into the ODD evaluation process. Similarly, consider the semantic dimension of a road's curvature where several ODDs exclude "sharp" curves. The definition of what constitutes a "sharp" curve can vary significantly between individuals, such as between an inexperienced driver and a racing driver. This variation occurs despite the availability of precise physical measures like the degree of curvature, which could objectively define sharpness and eliminate subjectivity.

The discrepancy highlights a broader issue with current publicly-available ODD specifications: they often lack sufficient granularity, leading to subjective interpretations of compliance criteria. This ambiguity and imprecision in ODDs complicates the application of formal methods and other forms of quality assurance, underscoring the need for rigorously defined ODDs that are robust against these variations. The current shortcomings of ODD descriptions for DASs underscore the difficulties in ensuring precise definition and adherence, pointing towards the necessity of approaches that provide utility by accommodating the nuanced realities of driving conditions.

B. Manual ODD Compliance Analysis of DAS Datasets

We now explore the degree to which DAS datasets conform to the ODD to determine whether it is appropriate for use in training and testing.

1) DAS and 3 Datasets: We primarily focus on comma.ai's openpilot ALC DAS and its NL ODD encompassing 11 semantic dimensions [9]; further discussion on the ODD dimensions is available in the online repository. We selected openpilot as it uses camera-based inputs to de-

[1] https://github.com/hildebrandt-carl/ODD_diLLMma_Artifact

TABLE I: Selected ODD semantic dimensions from NL description for comma.ai, GMC, and Tesla DASs.

Company	DAS	ODD Semantic Dimension									
		Weather and Environmental Factors						Roadway Characteristics			
		Heavy Rain	Rain	Sleet	Ice	Bright Light	Low Light	Sharp Curves	On-Off Ramps	Intersections	Traffic Signals
comma.ai [9]	ALC	×				×		×	×	×	
	ACC	×				×					×
GMC [18]	Super Cruise		×	×	×	×[a]		×[b]	×	×[c]	×
Tesla [19]	Traffic-Aware Cruise Control	×			×	×		×	×		
	Autosteer	×				×		×			
	Auto Lane Change	×						×			
	Full Self-Driving (Beta)		×			×	×		✓	✓	✓
	Autopark	×									
	Summon	×									
	Smart Summon	×									

[a] While Super Cruise does not mention Bright Light, it says "not when [...] there is too much glare". [b] Super Cruise says "not on winding [...] roads". [c] While Super Cruise does not mention intersections directly, it says "not on surface streets", excluding intersections.

TABLE II: ODD Compliance per Selected Data Subset

Subset	Dataset	In ODD	%	Out ODD	%
Both	comma.ai 2016	268	**56.6%**	232	46.4%
Both	comma.ai 2k19	288	**57.6%**	212	42.4%
Both	External JUtah	183	36.6%	317	**63.4%**
Pass	comma.ai 2016	191	**76.4%**	59	23.6%
Pass	comma.ai 2k19	222	**88.8%**	28	11.2%
Pass	External JUtah	137	**54.8%**	113	45.2%
Fail	comma.ai 2016	77	30.8%	173	**69.2%**
Fail	comma.ai 2k19	66	26.4%	184	**73.6%**
Fail	External JUtah	46	18.4%	204	**81.6%**

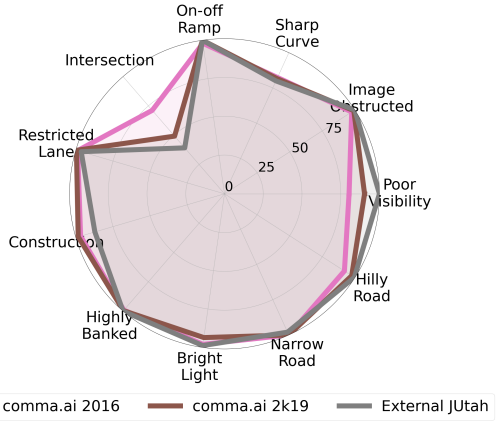


Fig. 2: Percentage Compliance with Semantic Dimensions Among Failing Images

termine its behavior, allowing us to leverage existing camera-based datasets. Further, openpilot is compatible with over 250 vehicle models [21] and has driven over 50 million miles while deployed [22] indicating the maturity of the system. Lastly, openpilot functions as is off-the-shelf, processing video inputs to generate steering angles.

Our dataset choices for evaluating comma.ai's openpilot system comprise the comma.ai 2016 dataset [23], which includes 11 videos totaling 7 hours, and the comma.ai 2k19 dataset [24], encompassing 2035 videos that amount to 34 hours. Given their source, we presumed these datasets would be compatible with the DAS's ODD. In addition, we incorporated 50 videos totalling 43 hours from the External JUtah collection [25], a curated global compilation of dashcam videos not affiliated with comma.ai, thus potentially diverging from the specified ODD. This diverse selection aims to fulfill three objectives: verify the efficacy of older datasets for training initial software versions, assess the relevance of newer datasets for advancing software capabilities, and evaluate model resilience using non-native datasets. Before their integration, all datasets were adjusted to meet the resolution and frame rate requirements of openpilot.

2) Selected Data for Analysis: To conduct the manual analysis we sample $1,500$ images from the datasets. The sampling technique is inspired by differential testing [26], comparing the steering angles produced by different DAS versions when given the same image to identify discrepancies indicative of failures, or similarities indicating passing. We implemented this approach by analyzing the outputs from three versions of the openpilot ALC [27], [28], [29]. When the maximum steering angle difference between any of these versions exceeded 45°, we interpreted this as a sign of failure in one of the versions—indicating that either an earlier version could not effectively handle the input, or the latest version exhibited a regression. Conversely, when all versions produced steering angles that differed by less than 1°, we considered this a strong indication of correct behavior, or a "pass." This process reduced the dataset from 4.6 million images to $691,369$. Under such conservative passing and failing definitions, 13% of images yielded a pass while 2% induced a failure. We randomly sampled 250 passing and 250 failing images from the reduced dataset across all videos to form 3 subsets of 500 images for manual investigation.

3) ODD Checking Process: Each researcher evaluated and marked each image according to its compliance to openpilot's ODD's 11 semantic dimensions (Figure 2) [9]. The researchers then met to discuss inconsistencies, developed rules to achieve consistency, and then re-reviewed all images; an overview of these rules is available in the online repository. The need for this consensus-building step highlights the difficulty in using ODDs specified in NL. Figure 3 highlights several examples of this challenge, dealing with the imprecision in defining "sharp curve" in Figure 3a, or unique cases such as a drive-thru in Figure 3d or hiking trail

(a) Imprecise: "Sharp Curve" (b) In-ODD (c) Out-ODD: Intersection (d) In-ODD but Likely Intended Out: Drive-Thru (e) In-ODD but Clearly Intended Out: Hiking Trail

Fig. 3: comma.ai 2k19 dataset images using openpilot ALC ODD (best viewed on a screen) illustrating annotation difficulties.

in Figure 3e which are likely out of the intended ODD but do not violate the specified ODD semantics.

4) ODD Compliance Results: Table II presents human-judged ODD compliance for each dataset, discriminated by images causing failures and passes. Contrary to expectations, nearly half of the sampled data from the company building openpilot, 46.4% and 42.4%, was found to be out of the ODD. Moreover, despite the expectation that failing images would predominantly fall outside the ODD—where correctness is not guaranteed—a substantial portion of the failing data, 30.8% and 26.4%, actually lies within the ODD. That means that openpilot would exhibit a significant difference in steering angle output of 45 degrees or more, depending on the version used, despite operating within the ODD.

For the External JUtah dataset, just more than one-third of the data is classified as in ODD, which is understandable considering that this dataset was not specifically collected for the openpilot system. Nonetheless, these findings illustrate the potential value of an approach which could automatically determine ODD compliance, revealing that large portions of already existing datasets—$178,199$ passing images and $11,046$ failing images, as extrapolated—could be effectively utilized for the training and testing of openpilot.

5) ODD Compliance per Semantic Dimension: We now examine the datasets' ODD compliance per semantic dimension to understand why images are out of ODD. As shown in Figure 2, the most common semantic dimension violated is "Intersection". This is understandable for External JUtah as it is an external dataset of mostly city driving images, but again we are surprised by the number of intersections in the comma.ai datasets. Similarly, it was strange to find basic dimensions such as "Poor Visibility" violated in 20% and 10% of comma.ai's inspected failing images respectively.

Findings: We find inconsistencies in NL ODD descriptions and compliance challenges. Analysis reveals that 46.4% and 42.4% of data from two DAS development datasets fall outside their ODDs. However, alternative datasets contain significant compliant data, with one offering an estimated $189,165$ new inputs for DAS development.

III. Approach

ODD-diLLMma analyzes real-world sensor data with respect to a DAS's ODD, producing a vector that encodes DAS compliance against the ODD semantic dimensions.

A. Overview

Figure 4 presents an overview of ODD-diLLMma. The inputs are a sensor dataset and the NL ODD specification. Each of the ODD semantic dimensions identified in the specification are reformulated, either manually or through an LLM-based Converter, as a yes-or-no question, e.g. "yes or no, was this taken at night". Then, all pairs of sensor inputs and questions are fed to the LLM-based Checker which generates an answer per pair. The yes-or-no answers generated by the LLM Checker are then converted to "inside" or "outside" of the ODD based on the phrasing of the question, i.e. answering "yes" to a question about an ODD exclusion results in "outside" for that semantic dimension. These responses are then encoded into a vector with one dimension for each semantic dimension. A vector of all "inside" indicates that the input did not violate the ODD, while all of the vectors with at least one "outside" illustrate a possible ODD non-compliance.

B. Large Language Models (LLMs)

ODD-diLLMma leverages LLMs to enable its analysis, building on their increasing potential to respond to a general array of inputs approaching human-level fidelity [8]. The field is rapidly progressing, with OpenAI, Google, and Meta each releasing LLMs in recent months [30], [31], [32]. LLMs take as input a *context* and a *prompt* and provide as output a *response* to the prompt. Advances in multi-modal LLMs have enabled handling of additional input and output types, such as images [8], [33], [34]. LLMs with multi-modal capabilities can, for example, take as context an image and, as a prompt, be asked to generate a caption for the image, which will be rendered as a textual output. We show next how ODD-diLLMma takes advantage of such emerging capabilities.

C. ODD-diLLMma Input and Conversion

ODD-diLLMma takes a set of ODD specifications written in NL and a sensor dataset collected on multiple scenes, where each scene includes the sensor readings (e.g., images, point clouds) that serve as *context* for the LLM Checker.

As discussed in Section II, *publicly available* ODDs are written in NL, often in the form of lists describing the ODD semantic dimensions. For our approach, we must convert the specifications into a structured format so the LLM Checker's responses can be unambiguously matched with the semantic dimensions. We transform the list of semantic dimensions into a series of yes-no questions due to prior demonstrated success in LLMs responding to this paradigm [35], [36]. This

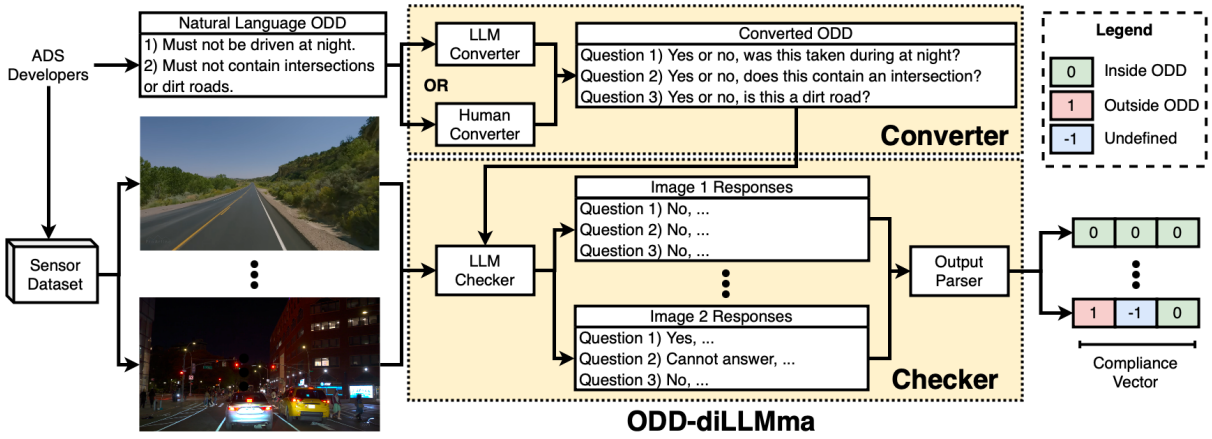


Fig. 4: ODD-diLLMma pipeline to judge sensor dataset compliance to a NL ODD description.

conversion is a one-time task that can be accomplished either manually or automatically, e.g. by an LLM-based Converter.

For example, a typical ODD specification might state: "Many factors can impact the performance of openpilot ALC and openpilot LDW, causing them to be unable to function as intended. These include, but are not limited to: Poor visibility (heavy rain, snow, fog, etc.) or weather conditions that may interfere with sensor operation..." [9]. This statement would be reformulated into a question like: "Yes or no, does the image exhibit poor visibility conditions such as heavy rain, snow, fog, or other weather conditions that may interfere with sensor operation?" By translating ODD specifications into this question format, we streamline the process for the LLM Checker to analyze sensor data in the context of these specifications while providing a specific and consistent interface for collecting data, enhancing the efficiency and effectiveness of our approach.

D. ODD-diLLMma Compliance Vector Generation

Given a sensor input, and a set of n translated ODD semantic dimension questions, our approach computes a single compliance vector describing how the sensor input complies with the ODD. Each of the n ODD semantic dimension questions are passed to the LLM Checker with the sensor input as context as shown in Figure 4.

The LLM Checker is prompted to output either "yes" or "no", which is then converted to "Inside ODD" (0), "Outside ODD" (1), or "Undefined" (-1). Given the inherent unpredictability of LLM outputs [37], we cannot guarantee that the LLM Checker will output explicitly and solely "yes" or "no". As such, we use a muti-method parsing strategy to interpret and validate the response. Strategies include looking directly for "yes" or "no", applying regular expressions to identify numbering patterns, and filtering out parts of the response based on context clues. The parsed responses across all dimensions are then concatenated to form the compliance vector which forms the basis of our analysis of datasets to judge their adherence to the DAS's ODD.

IV. Study

We address two questions on the efficacy of ODD-diLLMma in analyzing datasets concerning a DAS's ODD:

RQ1) Can ODD-diLLMma identify in-ODD failures?
RQ2) Does ODD-diLLMma accurately predict compliance overall and per semantic dimension?

A. Setup

The unit of analysis is comma.ai's open-source, road-tested DAS, openpilot, along with 11 dimensions of its ODD, checking compliance of 3 datasets totaling 85 hours of video.

1) LLM Checkers in ODD-diLLMma: As described in Section II-B, we first performed a human-annotated method on $1,500$ images over 11 ODD semantic dimensions to serve as a baseline to measure the proficiency of current LLMs.

We instantiated ODD-diLLMma with two off-the-shelf LLMs as checkers to explore multiple approaches for discerning semantic dimensions within images. One LLM comes from MiniGPT-4's [34] integration of the open-source Vicuna V0 13 billion parameter model [38]. The second LLM was OpenAI's proprietary ChatGPT-4V(ision) [33] accessed by their API. This allowed access to a more advanced model, but at the cost of ≈$0.02 per image-prompt pair.

Since prompt fine-tuning has been shown to enhance LLM performance, we used 10 known prompting strategies [39] and ChatGPT-4 to generate 10 alternate sets of questions. We then used all combinations of strategies and alternative questions to generate 100 unique prompts which we evaluated on a 10% sample of our dataset to find the prompt that yielded LLM responses that best matched the human annotations measured by F1-score [40], and called this technique Vicuna+. Conducting this prompt fine-tuning on ChatGPT-4V was not feasible due to the current daily rate limitations. Although this prompt fine-tuning allows further exploration of the potential of LLM performance, it requires additional calls to the LLM and available human annotations for portions of the dataset to identify the best prompt.

2) Converting the ODD: We structured each LLM Checker prompt into two key components: a *premise* for context and formatting, and *questions* that align with the ODD's semantic dimensions. The premise outlined the problem setup, which involved answering questions about an image taken from a car's front-facing camera. The questions were derived from the ODD descriptions on comma.ai's

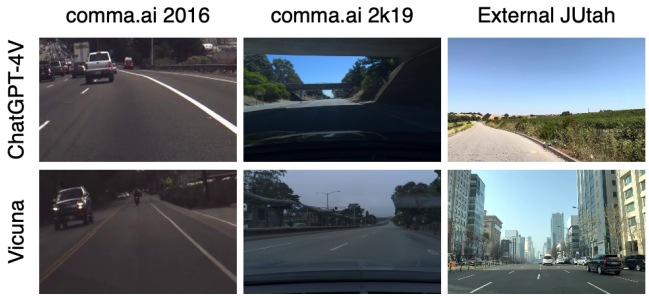


Fig. 5: Failures identified by ODD-diLLMma as in ODD.

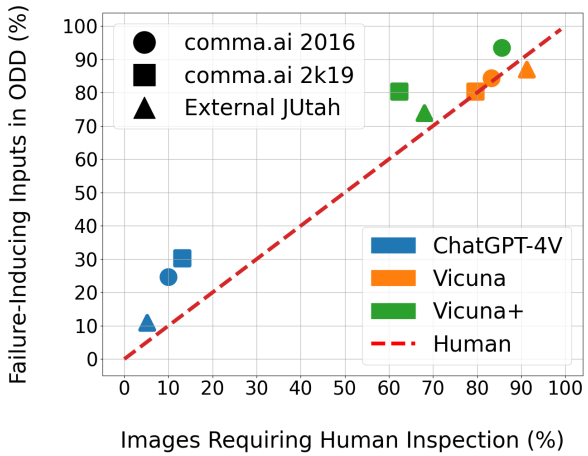


Fig. 6: In-ODD Failures vs Images Requiring Inspection

website [9], where we manually identified items in the ODD that could be assessed through an image. We then rephrased the ODD into structured yes-no questions. This process also involved breaking down complex ODD statements that encompassed multiple semantic dimensions. For example, a statement like "many factors can impact... [such as] sharp curves, on-off ramps, intersections" [9] actually contains three distinct scenarios: curves, on-off ramps, and intersections. Each of these was rephrased into a question, such as "Is the road we are driving on an intersection?" and included in the converted ODD. This manual translation was a one-time effort, resulting in a premise-question set prompt.

B. RQ1: Failures within the ODD

Consider a developer tasked with analyzing failure-inducing images to determine which ones are in ODD and thus should be further analyzed as they may represent a latent fault in the system. ODD-diLLMma is the first approach that can partially automate this process by selecting a subset of the failure-inducing images for the developer to review; ODD-diLLMma will provide the developer with a list of images judged to be in-ODD, and then the developer will review them to confirm. Thus, the first portion of the study compares the percentage of true in-ODD inputs found versus the percentage of inputs the developer reviewed. Figure 5 showcases failure-inducing images the LLM Checkers marked as in-ODD across the three datasets. In each case, we expect the DAS to perform well since the input is in-ODD, but it does not, highlighting the importance for the developers to be able to identify such cases.

Figure 6 shows the efficiency gains by automating the review process. The red dashed line represents the human annotation approach baseline, where each input image is manually reviewed. As the human examines a greater portion of the input set, we assume they would discover a proportionate amount of failure-inducing inputs that are within the ODD (reaching 100% when all images are inspected). Techniques that render scores above this line are more efficient as they allow the human to find more in-ODD failure-inducing inputs in the same amount of time. Techniques below this line are less efficient, requiring humans to spend more time analyzing failure-inducing inputs to find the same amount in ODD.

We observe several interesting patterns. First, ChatGPT-4V, across all datasets, is more efficient than the human annotation approach. At its peak on the comma.ai 2016 dataset, ChatGPT-4V correctly identifies 24.7% of all in-ODD failure-inducing inputs while requiring the human to review only 10.0% of images: $^{24.7\%}/_{10.0\%} = 2.47\times$ or 147% improvement in efficiency. By contrast, Vicuna consistently labels almost all inputs as in ODD, leading to efficiency on par or slightly below the baseline. However, Vicuna+ is able to improve upon this performance, showing efficiency over the baseline on all datasets. Vicuna+ achieves peak efficiency on comma.ai 2k19 finding 80.3% of failure-inducing inputs while only requiring the developer to review 62.4% of the dataset—an improvement of 28.7%. While a lesser gain in efficiency compared with ChatGPT-4V, Vicuna+ is able to identify a much larger quantity of in-ODD inputs.

The out-of-the-box success of the commercial ChatGPT-4V and the improvements shown by simple prompt fine-tuning on open-source models demonstrates the potential for ODD-diLLMma to provide utility in automating ODD compliance checking. Furthermore, while ODD-diLLMma is the first approach capable of identifying such failure-inducing inputs at a rate more efficient than humans, it is quite conservative by design. In order for a failure-inducing input to be marked in ODD, every semantic dimension must be compliant; this constraint could be relaxed to only consider partial ODD compliance. Additionally we note that the LLMs currently used have had no additional training for this problem domain, and we assume further improvements could be made through these methods.

> **RQ1 Findings:** ODD-diLLMma is the first approach capable of automatically identifying when failures are in ODD, achieving up to 147% improvement in efficiency when compared to purely manual analysis, enabling developers to find 24.7% of failure-causing images while only analyzing 10.0% of the dataset.

C. RQ2: ODD-diLLMma Accuracy

We analyze ODD-diLLMma's accuracy using different LLM Checkers across several dimensions and devising two

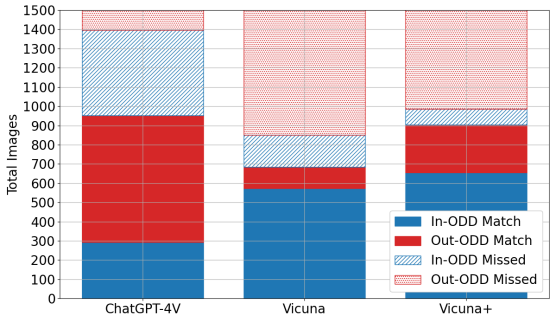


Fig. 7: ODD-diLLMma in-ODD Accuracy by LLM

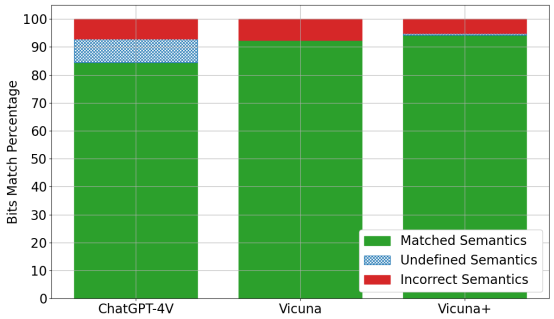


Fig. 8: ODD-diLLMma Semantic Accuracy by LLM

metrics. The first, in-ODD accuracy, measures if the LLM Checker correctly identified in/out of ODD status per image, with results in Figure 7. This evaluates the true and false positive rates for in/out-ODD labeling. The second, "semantic accuracy," assesses accuracy over the compliance vector's semantic dimensions, showing the LLM Checker's overall accuracy; results appear in Figure 8.

In Figure 7, "In-ODD Match" indicates true positive in-ODD, and "Out-ODD Missed" indicates false positive in-ODD, i.e. the LLM Checker mislabeled an out of ODD image as in-ODD. ChatGPT-4V appears conservative in its labeling of items in-ODD, achieving by far the lowest false positive in-ODD count and rate, while Vicuna and Vicuna+ are much less conservative, labeling many images as in-ODD. This helps to explain the RQ1 results, as efficiency is most increased when the in-ODD true positive to false positive ratio is high, i.e. it has high in-ODD precision. We find that ChatGPT-4V has a precision of $^{294}/_{(294+104)} = 73.9\%$ meaning that in 73.9% of cases where ChatGPT-4V says an image is in-ODD, it is correct. By comparison, Vicuna has a precision of $^{573}/_{(573+650)} = 46.9\%$, and Vicuna+ of $^{656}/_{(656+514)} = 56.1\%$. Despite ChatGPT-4V achieving the highest precision, Figure 8 shows that Vicuna and Vicuna+ achieve higher aggregate semantic accuracy with 84.4%, 92.6%, and 94.2% respectively. This performance difference is largely due to ChatGPT-4V's high number of "Undefined" answers where the model would say it was unsure; prompt refinements to encourage the model to take a stance may render improvements. Overall, the performance of ODD-diLLMma is encouraging at the semantic level.

Figure 9 further drills into this accuracy, showing the performance of each LLM checker per semantic dimension.

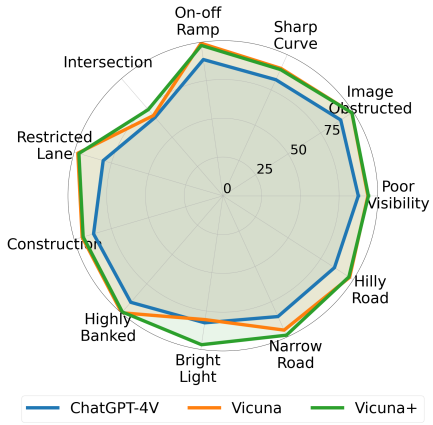


Fig. 9: ODD-diLLMma Accuracy per Semantic Dimension

On average, ODD-diLLMma achieves 90.3% accuracy, with a peak of 99.7% on the "Highly Banked" configuration. Notably, Vicuna and Vicuna+ demonstrate higher accuracy, again largely due to the high occurrence of "Undefined" responses by ChatGPT. Additionally, prompt fine-tuning significantly improved Vicuna's accuracy in the "Bright Light" dimension and, to a lesser extent, the "Intersection" dimension. These findings suggest potential avenues for further enhancements in the LLM Checkers within ODD-diLLMma.

RQ2 Findings: Building from RQ1, ODD-diLLMma demonstrates high precision in determining if an input is in ODD, with a maximum precision of 73.9%. ODD-diLLMma also achieves a high aggregate semantic accuracy of up to 94.2%. Further analysis shows accuracy is high across all semantic dimensions, achieving an average semantic dimension accuracy of 90.3% and a maximum accuracy of 99.7%.

V. Threats to Validity

Our study's external validity is affected by the choice of DAS and datasets. We chose comma.ai's system for its open-source nature and real-world use. Yet, each DAS behaves differently, and future studies with more DASs could address this. To mitigate dataset selection bias, we chose two comma.ai datasets and one with public dashcam footage. The internal validity of our study is largely dependent on the complexity of the components used, with LLMs being the most significant. LLMs have varying performance and are known to have hallucinations [37], both of which could impact the generality of our study. To capture a wide range of LLMs, we chose both an open-source and commercial model. While we used standardized prompt tuning across these models, certain uncontrollable factors, particularly with ChatGPT-4V's blackbox API, remain. Another threat is selection bias in our study due to the classification of 45-degree images as failing and 1-degree images as passing; while this threshold represents a significant steering difference, it may not fully capture true failures. Additionally, the low number of samples could further introduce bias. The human analysis

of ODD language descriptions and image compliance checks is subjective and potentially biased, though we mitigated this through the involvement of multiple participants. Nevertheless, we have made our code publicly available to ensure transparency in our methodology.

VI. Conclusion

This work highlights inconsistencies in today's publicly available NL ODD descriptions, demonstrating that multiple datasets used in development do not comply with their own ODDs. Furthermore, we illustrate the untapped potential of compliant data that remains unused or wasted due to the lack of automated compliance checks. We then introduced ODD-diLLMma, the first approach to automatically check if a dataset complies with a DAS's ODD. The approach leverages advances in LLMs to address the disconnect between ODD specifications and the datasets that underpin current DAS development. Our findings indicate that when instantiated with a sophisticated LLM, ODD-diLLMma makes the challenge tractable and can outperform the effectiveness of a purely manual approach. In the future, we will explore fine-tuning the LLMs to improve their performance, will carry out additional studies considering more DAS, datasets, and ODDs, and will develop analyses that can assist developers in closing the distance between their ODDs and their datasets.

References

[1] "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," April 2021. [Online]. Available: http://dx.doi.org/10.4271/J3016_202104

[2] T. Simonite, "Tesla Tests Self-Driving Functions with Secret Updates to Its Customers' Cars," 2016. [Online]. Available: https://www.technologyreview.com/2016/05/24/159976/tesla-tests-self-driving-functions-with-secret-updates-to-its-customers-cars/

[3] comma.ai Team, "Scaling for 10x user growth," https://blog.comma.ai/scaling-for-10x-user-growth/, 2021.

[4] W. Team, "Waymo significantly outperforms comparable human benchmarks over 7+ million miles of rider-only driving," https://waymo.com/blog/2023/12/waymo-significantly-outperforms-comparable-human-benchmarks-over-7-million/, 2023.

[5] S. Farm, "Cars with lights on in rain," https://www.flickr.com/photos/statefarm/16941638447, 2015.

[6] E. Christman, "Garmin dash cam picture," https://www.flickr.com/photos/gammaman/27993367996, 2016.

[7] Mattew, "A word to describe a curve and inaccurate road?" https://ell.stackexchange.com/questions/101107/a-word-to-describe-a-curve-and-inaccurate-road, 2016.

[8] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[9] commaai, "openpilot," https://github.com/commaai/openpilot/blob/b816b5b/docs/LIMITATIONS.md, 2023.

[10] L. Fraade-Blanar, M. S. Blumenthal, J. M. Anderson, and N. Kalra, *Measuring automated vehicle safety: Forging a framework*, 2018.

[11] P. Koopman and B. Osyk, "Safety argument considerations for public road testing of autonomous vehicles," *SAE International Journal of Advances and Current Practices in Mobility*, vol. 1, no. 2019-01-0123, pp. 512–523, 2019.

[12] H. Cho and R. J. Hansman, "Operational design domain (odd) framework for driver-automation systems," 2020.

[13] X. Zhang, S. Khastgir, and P. Jennings, "An odd-based scalable assurance framework for automated driving systems," SAE Technical Paper, Tech. Rep., 2023.

[14] I. Colwell, B. Phan, S. Saleem, R. Salay, and K. Czarnecki, "An automated vehicle safety concept based on runtime restriction of the operational design domain," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1910–1917.

[15] D. Z. für Luft-und Raumfahrt e. V. (DLR), "PEGASUS METHOD," 2019. [Online]. Available: https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/PEGASUS-Gesamtmethode.pdf

[16] E. Thorn, S. C. Kimmel, M. Chaka, B. A. Hamilton, *et al.*, "A framework for automated driving system testable cases and scenarios," United States. Department of Transportation. National Highway Traffic Safety Administration, Tech. Rep., 2018.

[17] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, "Survey on scenario-based safety assessment of automated vehicles," *IEEE access*, vol. 8, pp. 87 456–87 477, 2020.

[18] GMC, "Yukon/Yukon XL/Denali Owner's Manual," 2023. [Online]. Available: https://www.gmc.com/bypass/pcf/gma-content-api/resources/sites/GMA/content/staging/MANUALS/6000/MA6093/en_US/3.0/23_GMC_Yukon_Yukon_Denali_OM_en_US_U_16417394B_2022OCT10_2P.pdf

[19] Tesla, "Model Y Owner's Manual Software version: 2023.32 North America," 2023. [Online]. Available: https://www.tesla.com/ownersmanual/modely/en_us/Owners_Manual.pdf

[20] B. Kaiser, H. Weber, J. Hiller, and B. Engel, "Towards the definition of metrics for the assessment of operational design domains," *Open Research Europe*, vol. 3, 2023.

[21] comma.ai, "Openpilot supports 250+ vehicles," https://comma.ai/vehicles, 2023.

[22] comma.ai, "We're solving self driving cars while delivering shippable intermediaries," https://www.comma.ai/media, 2023.

[23] E. Santana and G. Hotz, "Learning a driving simulator," *arXiv preprint arXiv:1608.01230*, 2016.

[24] H. Schafer, E. Santana, A. Haden, and R. Biasini, "A commute in data: The comma2k19 dataset," *arXiv preprint arXiv:1812.05752*, 2018.

[25] JUtah, "Driving around the world, 30+ countries," https://www.youtube.com/@jutah, December 2023.

[26] W. M. McKeeman, "Differential testing for software," *Digital Technical Journal*, vol. 10, no. 1, pp. 100–107, 1998.

[27] comma.ai, "openpilot 5159878," https://github.com/commaai/openpilot/commit/515987838908c1a4f5c822919ccf2d78ebac144b, 2022.

[28] ——, "openpilot cb2a53a," https://github.com/commaai/openpilot/commit/cb2a53ae80ab3917986266290f37ef0228a6ca21, 2023.

[29] ——, "openpilot 2ebd7ab," https://github.com/commaai/openpilot/commit/2ebd7ab088ade61bbf661c140483fc477d444bc2, 2023.

[30] OpenAI, "GPT-4 Technical Report," 2023. [Online]. Available: https://arxiv.org/pdf/2303.08774.pdf

[31] S. Pichai, "An important next step on our AI journey," 2023. [Online]. Available: https://blog.google/technology/ai/bard-google-ai-search-updates/

[32] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[33] OpenAI, "GPT-4V(ision) System Card," 2023. [Online]. Available: https://cdn.openai.com/papers/GPTV_System_Card.pdf

[34] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *arXiv preprint arXiv:2304.10592*, 2023.

[35] Z. Rasool, S. Barnett, S. Kurniawan, S. Balugo, R. Vasa, C. Chesser, and A. Bahar-Fuchs, "Evaluating llms on document-based qa: Exact answer selection and numerical extraction using cogtale datase," *arXiv preprint arXiv:2311.07878*, 2023.

[36] H. Zhuang, Z. Qin, K. Hui, J. Wu, L. Yan, X. Wang, and M. Berdersky, "Beyond yes and no: Improving zero-shot llm rankers via scoring fine-grained relevance labels," *arXiv preprint arXiv:2310.14122*, 2023.

[37] Z. Bai, P. Wang, T. Xiao, T. He, Z. Han, Z. Zhang, and M. Z. Shou, "Hallucination of multimodal large language models: A survey," *arXiv preprint arXiv:2404.18930*, 2024.

[38] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, *et al.*, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

[39] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt, "A prompt pattern catalog to enhance prompt engineering with chatgpt," *arXiv preprint arXiv:2302.11382*, 2023.

[40] L. Wang, M. Han, X. Li, N. Zhang, and H. Cheng, "Review of classification methods on unbalanced data sets," *IEEE Access*, vol. 9, pp. 64 606–64 628, 2021.