Chat2Scenario: Scenario Extraction From Dataset Through Utilization of Large Language Model

Yongqi Zhao
Institute of Automotive Engineering
Graz University of Technology
Graz, Austria

Wenbo Xiao
Institute of Automotive Engineering
Graz University of Technology
Graz, Austria

Tomislav Mihalj
Institute of Automotive Engineering
Graz University of Technology
Graz, Austria

Jia Hu
College of Transportation Engineering
Tongji University
Shanghai, China

Arno Eichberger
Institute of Automotive Engineering
Graz University of Technology
Graz, Austria

***Abstract*—The advent of Large Language Models (LLM) provides new insights to validate Automated Driving Systems (ADS). In the herein-introduced work, a novel approach to extracting scenarios from naturalistic driving datasets is presented. A framework called Chat2Scenario is proposed leveraging the advanced Natural Language Processing (NLP) capabilities of LLM to understand and identify different driving scenarios. By inputting descriptive texts of driving conditions and specifying the criticality metric thresholds, the framework efficiently searches for desired scenarios and converts them into ASAM OpenSCENARIO[1] and IPG CarMaker text files[2]. This methodology streamlines the scenario extraction process and enhances efficiency. Simulations are executed to validate the efficiency of the approach. The framework is presented based on a user-friendly web app and is accessible via the following link: https://github.com/ftgTUGraz/Chat2Scenario.**

***Index Terms*—Large Language Model, Scenario Extraction, Automated Driving Systems, Virtual Testing**

I. Introduction

It has been proven that mileage-based on-road testing is not sufficient for the validation of ADS, as Automated Vehicles (AV) must be driven billions of miles to demonstrate their reliability [1]. To increase testing efficiency, a scenario-based method was proposed in project PEGASUS[3] aiming to expose ADS in virtual driving environments derived from the real world. However, this approach heavily relies on the measurements of real-world traffic and high-fidelity simulation platforms. Adequate measurement data of real-world traffic ensures a reliable data source; simulators provide an efficient alternative to guarantee safety, as long as it is close to reality.

In recent years, there have been intensive investigations into contributions related to high-quality, cost-effective dataset provision (cf. [2]–[10]) and the release of portable simulation platforms (cf. [11]–[13]). However, the availability of portable and publicly accessible automation tools for reconstructing the measurements within these simulation platforms has been limited. Karunakaran et al. [14] developed a tool for identifying and extracting lane change scenarios from LiDAR point clouds. Zhu et al. [15] proposed a framework for extracting ADS disengagement scenarios from AV road testing data. Montanari et al. [16], [17] created a tool to extract concrete scenarios from test vehicle bus communication data based on maneuvers. Zhang et al. [18] introduced a toolkit to extract accident scenarios from traffic surveillance videos. However, these tools face several issues: 1) compatibility is limited to datasets that are either difficult to acquire at scale, not publicly accessible, or require extensive pre-processing; 2) the ability to extract only one type of scenario, resulting in a limited scope; 3) the inability to quantitatively evaluate the criticality of the generated scenarios; 4) the lack of open-source availability or user-friendly interfaces, making practical application laborious.

Based on the aforementioned issues, the Chat2Scenario framework is introduced. This framework incorporates the latest OpenAI LLM, *gpt-4-1106-preview*[4], to extract concrete scenarios from naturalistic driving datasets. In this work, the highD dataset (refer to [2]), comprising vehicle trajectories on German highways collected via drones, is employed. This approach is advantageous not only because addresses the issue of limited data sources but also due to the minimal requirement for pre-processing. Datasets in the same format from various locations (e.g., intersection [4], roundabout [5],

This work was supported by the National Key R&D Program of China under Grant Nr. 2022YFE0117100, and by the FFG in the research project PECOP (FFG Projektnummer 893988), as part of the “Bilateral Cooperation Austria - People’s Republic of China / MOST 2nd Call” program. *Corresponding author: Wenbo Xiao (wenbo.xiao@student.tugraz.at)*

[1] https://www.asam.net/standards/detail/openscenario/

[2] https://ipg-automotive.com/de/support/supportanfrage/faq/usage-of-user-inputs-from-a-file-in-a-maneuver-133/

[3] https://www.pegasusprojekt.de/en/home

[4] https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo

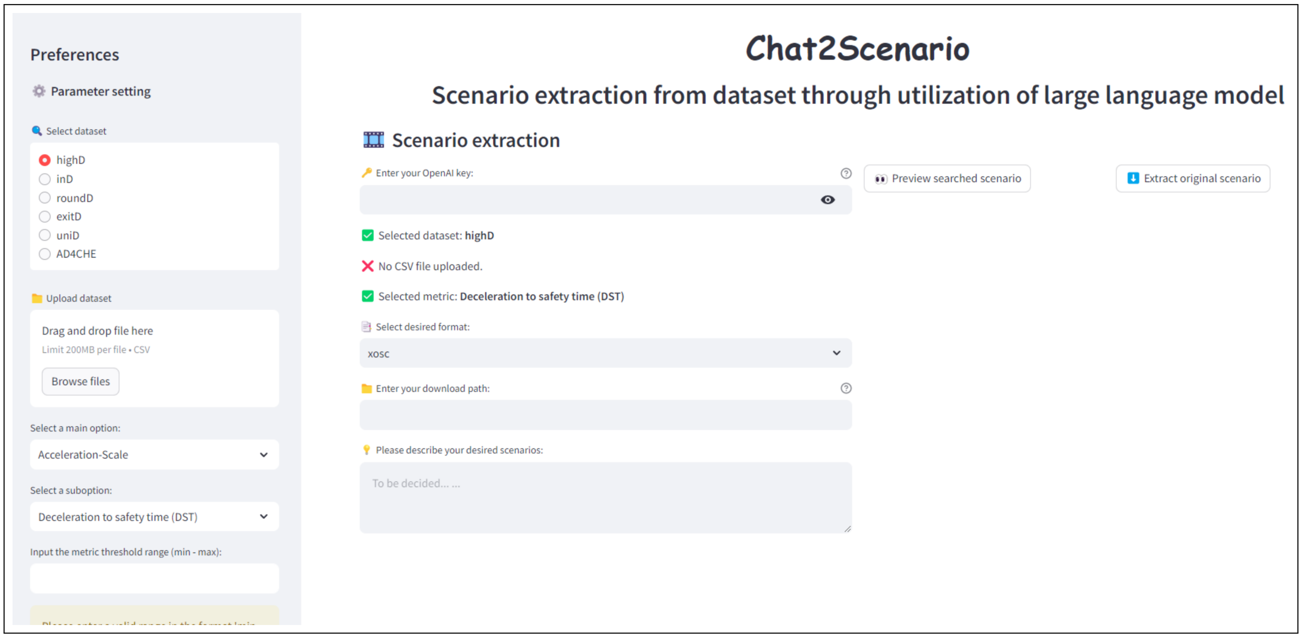


Fig. 1: Overview of Chat2Scenario web app.

highway exits [10], university campus[5]) are also publicly available and can be potentially integrated into this tool.

Subsequently, the OpenAI LLM is leveraged to interpret scenarios described in natural language, thereby broadening the range of searchable scenario types. Next, several metrics are provided to quantify the criticality of generated scenarios. Finally, the tool is delivered as a user-friendly and practical web app (see Fig. 1) to enhance usability. The contributions of this work are summarized as follows:

1) The OpenAI LLM is utilized to enhance scenario searching efficiency and expand the searchable scenario types.
2) Criticality metrics-based scenario filtering criterion is provided to promote the searching accuracy.
3) A practical and shareable web app is released to connect the naturalistic driving dataset and the simulation platform for ADS validation.

The presented framework would be useful to facilitate the process of the ADS function test. The outcome should also provide new insight for the ADS testing engineers to simplify the search and analysis of complex datasets.

II. Terminology and Dataset Format

A. Definition of Activity and Event

An *Activity* is defined as *the minimal unit in a scenario's dynamics, representing the temporal progression of state variables, where its end signifies the commencement of the subsequent activity* [19]. An *Event* indicates *the time instant when a transition of state occurs, such that before and after an event, the state corresponds to two different activities* [19]. The concepts of *Event* and *Activity* are visualized in Fig. 2.

[5]https://levelxdata.com/unid-dataset/

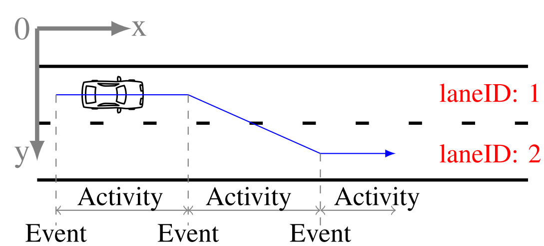


Fig. 2: Visualization of *Event* and *Activity*: blue arrow represents the vehicle trajectory [2], [19].

B. Dataset Format

The highD dataset comprises multiple recordings, each encapsulated within a CSV file. Each file encompasses a suite of vehicle trajectories, providing comprehensive details such as the data frame, vehicle ID, position, velocity, acceleration, and the current lane ID for each respective trajectory, as depicted in Tab. I. The global coordinate system's origin is positioned at the upper left corner, with the horizontal and vertical axes defined as the x-axis and y-axis, respectively. Lanes within this system are sequentially numbered starting from 1, as depicted in Fig. 2.

TABLE I: Available information in highD dataset [2]

Name	Unit	Name	Unit
frame	[-]	xVelocity	[m/s]
id	[-]	yVelocity	[m/s]
x	[m]	xAcceleration	[m/s^2]
y	[m]	yAcceleration	[m/s^2]
width	[m]	laneId	[-]
height	[m]	...	...

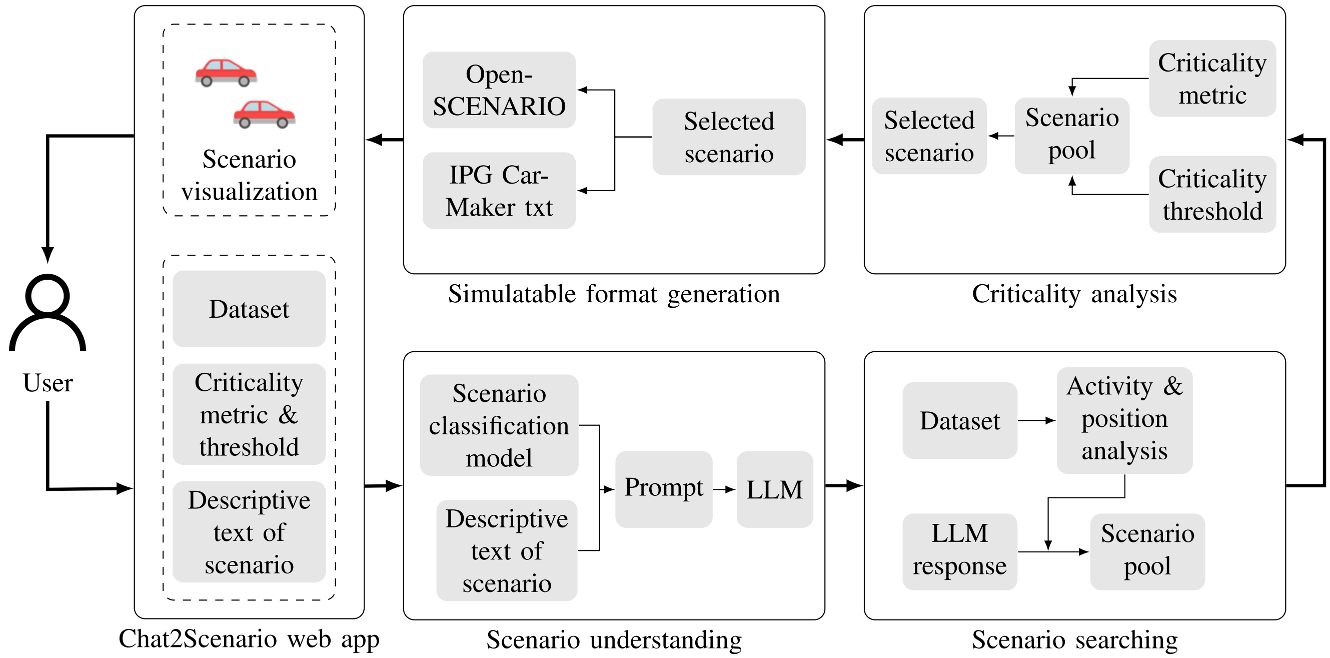


Fig. 3: Schematic overview of the Chat2Scenario framework operations.

III. Methodology

In Fig. 3, it is illustrated the workflow of Chat2Scenario system. This process begins with the user uploading a dataset to the Chat2Scenario web app and entering a scenario description along with the criticality metric and threshold. LLM subsequently interprets the descriptive text of the scenario. Based on the LLM response, scenarios matching the requirements are added to the scenario pool, where their criticality is analyzed. Scenarios not meeting the threshold criteria are excluded. Finally, the selected scenarios are converted into ASAM OpenSCENARIO and IPG CarMaker text. These scenarios can be visualized via the web app. Each module in the flowchart is detailed in this section.

A. Chat2Scenario Web App

The shareable web app depicted in Fig. 1 is developed using the Python programming language and leverages the *Streamlit*[6] library. It offers a user-friendly interface that enables users to configure parameters and visually explore selected scenarios with ease.

B. Scenario Understanding

In the scenario understanding module, the NLP capabilities of LLM are utilized to identify and categorize the dynamic behaviors and positions of vehicles within driving scenarios. A key factor in this process is the strategic prompt engineering, which directs the LLM to generate precise and relevant responses. In this study, the prompts provided to the LLM integrate a scenario classification model. This integration allows the LLM to interpret scenario narratives and accurately align semantic labels with a well-structured framework, transforming unstructured text into structured scenario data. Further details on the scenario classification model and the interface and the prompt engineering are elaborated in subsequent sections.

1) Scenario Classification Model: In Fig. 4, the classification model for highway traffic scenarios is illustrated. This model categorizes the information into two primary sections: vehicle activity and target vehicle's position w.r.t. the ego vehicle. Vehicle activity is subdivided into longitudinal and lateral activities. Longitudinal activity pertains to velocity, with three possible states: *Keep velocity*, *Acceleration*, or *Deceleration* [20]. Lateral activity relates to the vehicle's interaction with traffic lanes, comprising *Follow lane*, *Lane change left*, or *Lane change right*.

The relative position of the target vehicle concerning the ego vehicle is crucial for resolving ambiguities in scenarios. For instance, consider an unspecified scenario where *the ego vehicle follows the lane, and a target vehicle changes lane to the right*. This situation could correspond to any scenarios depicted in Fig. 5a - 5c. However, if it is specified that *the target vehicle begins in the left adjacent lane and ends up in front of the ego vehicle within the same lane*, only the scenario in Fig. 5c satisfies these conditions.

Target vehicles position w.r.t. ego vehicle is categorized as being in the *Same Lane*, *Adjacent Lane*, or *Lane Next to Adjacent Lane*. The presence of vehicles in the same or adjacent lanes is significant for the decision-making of the AV and is thus included [21]. Vehicles in the lane next to the adjacent lane are also considered due to their potential to merge into the ego vehicle's lane [21]. Vehicles in other lanes are excluded from consideration as they are unlikely to interact with the ego vehicle. Regarding the *Same Lane*, target vehicles are either *Behind* or in *Front* of the ego vehicle.

[6]https://docs.streamlit.io/library

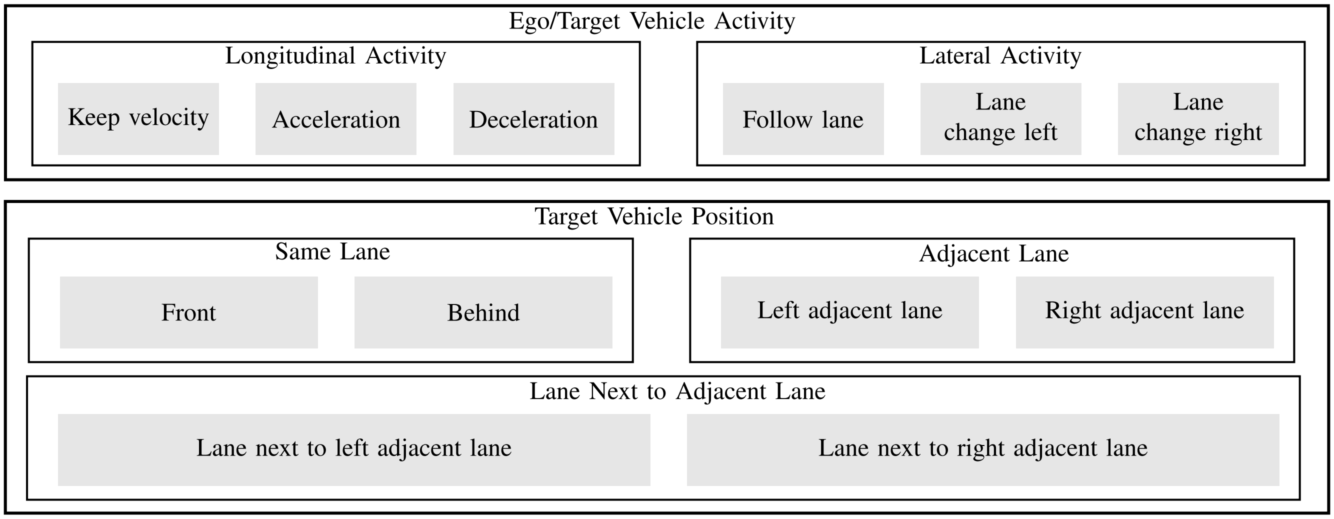
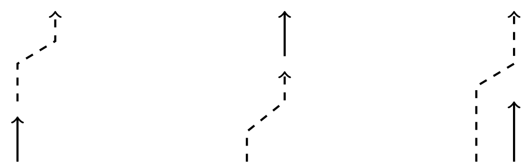


Fig. 4: Scenario classification model for highD traffic.



(a) (b) (c)

Fig. 5: Visualization of scenarios: dashed and solid lines represent target and ego vehicle trajectories respectively.

As for the *Adjacent lane*, the target vehicles are situated in the *Left* or *Right adjacent lane*. The categorization is analogous for vehicles in the *Lane Next to Adjacent Lane*.

2) Prompt Engineering of LLM: Prompt engineering refers to the strategic formulation of input queries to effectively guide the LLM's responses, optimizing for more accurate, relevant, and useful outputs. Informed by the six strategies from OpenAI's prompt engineering guide[7], a structured prompt for LLM optimization is proposed, as depicted in Figure 6. The prompt consists of five segments. The first segment, Fig. 6a, delineates the role of the LLM as an advanced AI tool for scenario analysis, specially tasked with interpreting driving scenarios following a pre-established classification model shown in Fig. 4. This structured approach is designed to minimize response variability, thereby simplifying the subsequent processing.

In Fig. 6b, the user is required to provide a detailed description of the driving scenario as '**descriptive_text_of_scenario**'. This input serves as the contextual basis for the LLM's analytical extraction.

The details of the task are articulated in Fig. 6c, which sets forth the expected structured response from the LLM. It is necessary to use curly braces '{}' and square brackets '[]' to systematically encapsulate the identified attributes of the scenario. In instances involving several target vehicles, the response employs a numbering system with '#' and a sequential numeral to organize the data. This structured format is imperative for the automated parsing of scenario data, thus optimizing programming efficiency.

[7]https://platform.openai.com/docs/guides/prompt-engineering

An application of this framework is demonstrated in Fig. 6d, providing an example of how the LLM should process a specific driving situation. It depicts an instance in which the ego vehicle maintains its velocity and lane position while another vehicle executes lane changes and accelerations. The subsequent subprompt in Fig. 6e reinforces the analytical task, guaranteeing precision and uniformity in the results.

C. Scenario Searching

The primary objective of scenario searching is to evaluate the congruence between vehicle trajectories in the datasets and the LLM's responses. This congruence assessment hinges on the identified activities of both the ego and target vehicles, as well as their relative positional relationships. The methodologies are expounded upon in the following sections.

1) Activity Identification: The distinction between subcategories of longitudinal activities A_{lon} depends on the comparison of longitudinal acceleration a_{lon} with a predefined acceleration threshold $a_{\text{lon}}^{\text{thr}}$ [22]. This comparison must satisfy the following equation:

$$A_{\text{lon}}(a_{\text{lon}}) = \begin{cases} \textit{Deceleration}, & a_{\text{lon}} < -a_{\text{lon}}^{\text{thr}}, \\ \textit{Acceleration}, & a_{\text{lon}} > a_{\text{lon}}^{\text{thr}}, \\ \textit{Keep velocity}, & \text{otherwise.} \end{cases} \quad (1)$$

The variation in lane ID, denoted as (ΔL), indicates lateral activities A_{lat} and helps in identifying lane changes. Additionally, the vehicle's longitudinal velocity (v_{lon}), in relation to the x-axis, determines the direction of the change,

(a) System, you are an AI trained to understand and classify driving scenarios based on specific frameworks. Your task is to analyze the following driving scenario and classify the behavior of both the ego vehicle and the target vehicle according to the given classification framework. Please follow the framework strictly and provide precise and clear classifications. The framework is as follows: **scenario_classification_model**

(b) Scenario Description: **descriptive_text_of_scenario**

(c) Provide a detailed classification for both the ego vehicle and the target vehicle(s). The response should be formatted exactly as shown in this structure:

```
{
  Ego Vehicle: {Ego longitudinal activity: ['Your Classification'], Ego lateral activity: ['Your Classification']},
  Target Vehicle #1:
  {
      Target start position: {'Your Classification': ['Your Classification']},
      Target end position: {'Your Classification': ['Your Classification']},
      Target behavior: {target longitudinal activity: ['Your Classification'],
                        target lateral activity: ['Your Classification']'}
  }
  Target Vehicle #2:
  {
    ......
    ......
  }
}
```

(d) Example: If an ego vehicle is maintaining speed and following its lane, while another vehicle is initially in the left adjacent lane and is accelerating, then changing lanes to the right; finally driving on the front of ego vehicle, the classification would be:

```
{
  Ego Vehicle: {Ego longitudinal activity: ['keep velocity'], Ego lateral activity: ['follow lane']},
  Target Vehicle:
  {
      Target start position: {'adjacent lane': ['left adjacent lane']},
      Target end position: {'same lane': ['front']},
      Target behavior: {target longitudinal activity: ['acceleration'],
                        target lateral activity: ['lane change right']'}
  }
```

(e) Remember to analyze carefully and provide the classification as per the structure given above.

Fig. 6: Description prompt submitted to LLM.

whether left or right. This is quantified by the following equation:

$$A_{\text{lat}} = \begin{cases} \textit{Follow lane}, & \Delta L = 0, \\ \textit{Lane change right}, & (\Delta L > 0 \text{ and } v_{\text{lon}} > 0) \text{ or} \\ & (\Delta L < 0 \text{ and } v_{\text{lon}} < 0), \\ \textit{Lane change left}, & (\Delta L < 0 \text{ and } v_{\text{lon}} > 0) \text{ or} \\ & (\Delta L > 0 \text{ and } v_{\text{lon}} < 0). \end{cases} \tag{2}$$

2) Position Identification: The relative position of the target vehicle w.r.t. the ego vehicle, denoted as $P_{\text{tgt}}^{\text{ego}}$, can be determined by the absolute value of the lane ID difference between the target and ego vehicles, given by $\|\Delta L_{\text{ego}}^{\text{tgt}}\|$, which is the modulus of the difference in lane IDs, $\|L_{\text{tgt}} - L_{\text{ego}}\|$, and the position difference along the x-axis, Δx.

For situations where $v_{\text{lon}} < 0$, the position $P_{\text{tgt}}^{\text{ego}}$ should satisfy the following conditions:

$$P_{\text{tgt}}^{\text{ego}} = \begin{cases} \textit{Front}, & \|\Delta L_{\text{ego}}^{\text{tgt}}\| = 0 \text{ and } \Delta x < 0, \\ \textit{Behind}, & \|\Delta L_{\text{ego}}^{\text{tgt}}\| = 0 \text{ and } \Delta x > 0, \\ \textit{Left adjacent lane}, & \Delta L_{\text{ego}}^{\text{tgt}} = 1, \\ \textit{Right adjacent lane}, & \Delta L_{\text{ego}}^{\text{tgt}} = -1, \\ \textit{Lane next to the left adjacent lane}, & \Delta L_{\text{ego}}^{\text{tgt}} = 2, \\ \textit{Lane next to the right adjacent lane}, & \Delta L_{\text{ego}}^{\text{tgt}} = -2. \end{cases} \tag{3}$$

Similarly, when $v_{\text{lon}} > 0$, the position $P_{\text{tgt}}^{\text{ego}}$ is defined by:

$$P_{\text{tgt}}^{\text{ego}} = \begin{cases} \textit{Front}, & \|\Delta L_{\text{ego}}^{\text{tgt}}\| = 0 \text{ and } \Delta x > 0, \\ \textit{Behind}, & \|\Delta L_{\text{ego}}^{\text{tgt}}\| = 0 \text{ and } \Delta x < 0, \\ \textit{Left adjacent lane}, & \Delta L_{\text{ego}}^{\text{tgt}} = -1, \\ \textit{Right adjacent lane}, & \Delta L_{\text{ego}}^{\text{tgt}} = 1, \\ \textit{Lane next to the} & \\ \textit{left adjacent lane}, & \Delta L_{\text{ego}}^{\text{tgt}} = -2, \\ \textit{Lane next to the} & \\ \textit{right adjacent lane}, & \Delta L_{\text{ego}}^{\text{tgt}} = 2. \end{cases} \tag{4}$$

D. Criticality Analysis

The scenario pool may encompass a multitude of scenarios that correspond to the input descriptive text. To quantitatively ascertain the scenarios most pertinent to the testing task, their criticalities must be evaluated. Consequently, this study incorporates some metrics summarized by Westhofen et al. [23] for assessing scenario criticality.

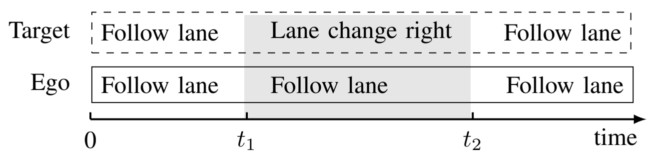


Fig. 7: Illustration of criticality analysis within a scenario: the gray area denotes the segment under analysis.

A further consideration is determining which scenes within a scenario warrant criticality analysis. In this work, criticality is computed exclusively for the scenario segments where both the ego and target vehicles' activities and their relative positions are satisfied. For example, in the scenario depicted in Fig. 5c, criticality is assessed from t_1 to t_2, as shown in Fig. 7. The criticality metrics utilized in Chat2Scenario are cataloged in Table II.

TABLE II: Criticality Metric in Chat2Scenario [23]

Metric Category	Metric Name
Acceleration-Scale	Deceleration to safety time (DST) Required longitudinal acceleration (RLongA) Required lateral acceleration (RLatA) Required acceleration (RA)
Distance-Scale	Proportion of stopping distance (PSD) Distance headway (DHW)
Jerk-Scale	Longitudinal jerk (LongJ) Lateral jerk (LatJ)
Time-Scale	Encroachment time (ET) Post-encroachment time (PET) Time to collision (TTC) Potential time to collision (PTTC) Time exposed TTC (TET) Time integrated TTC (TIT) Time to closest encounter (TTCE) Time to brake (TTB) Time to kickdown (TTK) Time to steer (TTS) Time headway (THW)
Velocity-Scale	Δv

E. Simulatable Format Generation

The Chat2Scenario platform facilitates the generation of scenarios in two formats: ASAM OpenSCENARIO and IPG CarMaker text.

1) ASAM OpenSCENARIO: The OpenSCENARIO files, generated using the *scenariogeneration*[8] Python package, are compatible with multiple simulators, including Esmini [13] and CARLA [11]. The XML schema is utilized for defining the scenarios in OpenSCENARIO. In this work, scenarios are reconstructed in Esmini based on vehicular trajectories. This process involves specifying a Vertex for each timestamp, illustrated in Fig. 8. For each Vertex, the vehicle's WorldPosition is delineated, with coordinates (x, y, z) derived directly from the dataset. Regarding the attitude, the heading angle (h) is constrained to 0 or π radians, reflecting the road's alignment with the x-axis. Both pitch (p) and roll (r) angles are set to 0 radians by default.

```
<Vertex time="0.0">
    <Position>
        <WorldPosition x="389.16" y="-14.27" z="0.0"
h="0.0" p="0.0" r="0.0"/>
    </Position>
</Vertex>
```

Fig. 8: An exemplary vertex in OpenSCENARIO.

2) IPG CarMaker Text: The IPG CarMaker text format is tailored to incorporate real-world measurements of vehicle maneuvers into the CarMaker simulation environment. As depicted in Fig. 9, the format begins with a timestamp in the first column. Successive columns record the vehicle's global position coordinates - longitudinal (x) and lateral (y) positions. The initial row captures the vehicle's position at the scene's start, with a timestamp of zero and "162" as the vehicle's identifier. In scenarios with multiple vehicles, their data are adjacently aligned, employing unique identifiers for each. Notably, this format is applied solely to non-ego vehicles. Regarding the ego vehicle, the trajectory is defined on the road network through the edition of UserPath.Nodes. More details are available in the IPGRoad document [24].

```
#time,   x_162,    y_162,    ...
 0.0,    389.16,   -14.27,   ...
 ...     ...       ...       ...
```

Fig. 9: Illustration of IPG CarMaker text format.

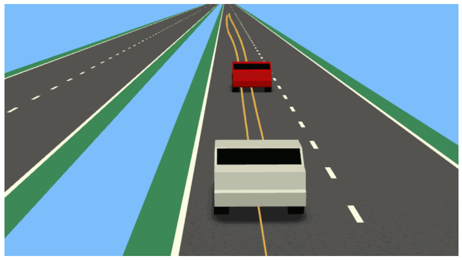
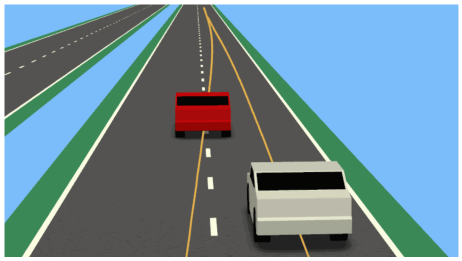
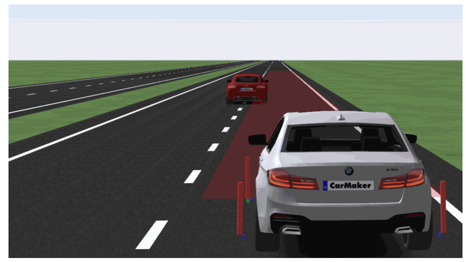
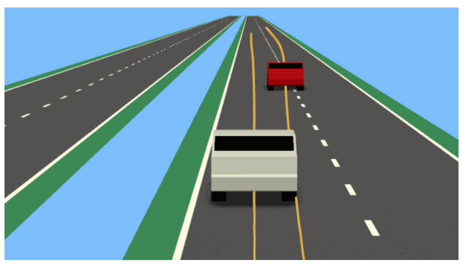
IV. Result and Discussion

A. Qualitative Evaluation

The framework Chat2Scenario is validated through the extraction of three typical driving scenarios: following, cut-in, and cut-out scenarios. These scenarios are pivotal in testing ADS due to their frequency of occurrence and potential risk in daily driving. The following scenario tests the ADS

[8] https://github.com/pyoscx/scenariogeneration

TABLE III: Exemplary extracted scenarios through the utilization of Chat2Scenario

	Descriptive Text	Extracted Scenario (Esmini)	Extracted Scenario (CarMaker)
Following	The ego vehicle follows the lane and decelerates. Target vehicle #1, which is in front of the ego vehicle in the same lane, also decelerates.		
Cut-in	The ego vehicle maintains its lane and velocity. Initially, Target Vehicle #1 is driving in the left adjacent lane. It then accelerates and changes lanes to the right, eventually driving in front of the ego vehicle.		
Cut-out	The ego vehicle follows the lane and maintains its velocity. Target vehicle #1, initially driving in front of the ego vehicle in the same lane, accelerates and changes lanes to the right.		

capability to maintain safe following distances and respond to varying speeds of traffic. The cut-in and cut-out scenarios are critical for evaluating an ADS's lane-changing and overtaking strategies, as they involve the ego vehicle's reaction to other vehicles. These scenarios are successfully extracted from the dataset and reconstructed in Esmini and CarMaker, as detailed in Tab.III.

B. Quantitative Evaluation

In the highD dataset, ground truth labels for various scenarios are absent. To address this gap, semantic labels are manually generated through an analysis facilitated by a MATLAB-based visualization tool[9]. Given the intensive nature of the work, only one file - track #36, which lasts about 27 minutes - is randomly selected from a total of 60 files for detailed analysis. In the labeling process, human driving experiences are used, but the vehicle's longitudinal activities are excluded due to the difficulty of manual identification from animated data. The quality of these labels is further verified by an independent reviewer. The effectiveness of the Chat2Scenario tool is assessed by comparing its outputs with the ground truth of track #36, and the results are presented in Tab. IV.

[9]https://github.com/RobertKrajewski/highD-dataset

For the "following" scenario, a promising level of precision is demonstrated; however, this is mitigated by a significant rate of "false negative", which can be attributed to the exclusion of scenarios that fall below the predefined duration threshold. In terms of the "cut-in" and "cut-out" scenarios, the values across all metrics indicate a robust competence in effectively identifying these scenarios. Overall, the capabilities of Chat2Scenario for scenario identification are substantiated by the quantitative metrics presented in the table.

V. Conclusion and Future Work

This paper has introduced Chat2Scenario, a publicly accessible web app that advances the extraction of concrete scenarios from naturalistic driving datasets. The platform interprets scenario descriptions in natural language and evaluates their criticality with precision, thereby streamlining the scenario generation process. The validity and the practicality of Chat2Scenario are substantiated through simulations in Esmini and CarMaker.

Comprehensive validation of the framework is to be primarily focused upon in future work to ensure its robustness and reliability. Additionally, the expansion of dataset diversity and the refinement of criticality metrics for customized evaluation are also planned.

TABLE IV: Quantitative evaluation of Chat2Scenario in track #36

Scenario Category	True Positive	False Positive	False Negative	Accuracy	Precision	Recall	F1 Score
following	2479	15	814	0.749	0.994	0.752	0.857
cut-in	248	23	39	0.800	0.915	0.864	0.889
cut-out	265	15	32	0.849	0.946	0.892	0.919

ACKNOWLEDGMENT

The authors would like to thank Dr. Hexuan Li for the discussion during the conception phase, Faris Orucevic for the technical support about CarMaker, and Zhengyu Wang for generating the ground truth labels.

REFERENCES

[1] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?," *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.

[2] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2118–2125, 2018.

[3] Y. Zhang, C. Wang, R. Yu, L. Wang, W. Quan, Y. Gao, and P. Li, "The ad4che dataset and its application in typical congestion scenarios of traffic jam pilot systems," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 5, pp. 3312–3323, 2023.

[4] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1929–1934, 2020.

[5] R. Krajewski, T. Moers, J. Bock, L. Vater, and L. Eckstein, "The round dataset: A drone dataset of road user trajectories at roundabouts in germany," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, 2020.

[6] B. Coifman and L. Li, "A critical evaluation of the next generation simulation (ngsim) vehicle trajectory dataset," *Transportation Research Part B: Methodological*, vol. 105, pp. 362–377, 2017.

[7] P. Spannaus, P. Zechel, and K. Lenz, "Automatum data: Drone-based highway dataset for the development and validation of automated driving software for research and commercial applications," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1372–1377, 2021.

[8] E. Barmpounakis and N. Geroliminis, "On the new era of urban traffic monitoring with massive drone data: The pneuma large-scale field experiment," *Transportation Research Part C: Emerging Technologies*, vol. 111, pp. 50–71, 2020.

[9] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pp. 549–565, Springer, 2016.

[10] T. Moers, L. Vater, R. Krajewski, J. Bock, A. Zlocki, and L. Eckstein, "The exid dataset: A real-world trajectory dataset of highly interactive highway scenarios in germany," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, pp. 958–964, 2022.

[11] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, vol. 78 of *Proceedings of Machine Learning Research*, pp. 1–16, PMLR, 13–15 Nov 2017.

[12] IPG Automotive GmbH, "Carmaker." https://ipg-automotive.com/en/products-solutions/software/carmaker/, 2023. Accessed: 2023-12-20.

[13] "Environmental simulator minimalistic (esmini)," 2018.

[14] D. Karunakaran, J. S. Berrio, S. Worrall, and E. Nebot, "Automatic lane change scenario extraction and generation of scenarios in openx format from real-world data," *arXiv preprint arXiv:2203.07521*, 2022.

[15] Z. Zhu, R. Philipp, Y. Zhao, C. Hungar, J. Pannek, and F. Howar, "Automatic disengagement scenario reconstruction based on urban test drives of automated vehicles," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–8, IEEE, 2023.

[16] F. Montanari, C. Stadler, J. Sichermann, R. German, and A. Djanatliev, "Maneuver-based resimulation of driving scenarios based on real driving data," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1124–1131, 2021.

[17] F. Montanari, Y. A. Akkaya, N. Boßmann, J. Sichermann, M. Müller, A. J. Aigner, and D. D'Sa, "OSC-Generator."

[18] Z. Xinxin, L. Fei, and W. Xiangbin, "Csg: Critical scenario generation from real traffic accidents," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1330–1336, IEEE, 2020.

[19] H. Elrofai, J.-P. Paardekooper, E. de Gelder, S. Kalisvaart, and O. O. den Camp, "Scenario-based safety validation of connected and automated driving," *Netherlands Organization for Applied Scientific Research, TNO, Tech. Rep*, 2018.

[20] L. Hartjen, R. Philipp, F. Schuldt, B. Friedrich, and F. Howar, "Classification of driving maneuvers in urban traffic for parametrization of test scenarios," in *9. Tagung Automatisiertes Fahren*, 2019.

[21] R. Philipp, J. Rehbein, F. Grün, L. Hartjen, Z. Zhu, F. Schuldt, and F. Howar, "Systematization of relevant road users for the evaluation of autonomous vehicle perception," in *2022 IEEE International Systems Conference (SysCon)*, pp. 1–8, 2022.

[22] P. Bokare and A. Maurya, "Acceleration-deceleration behaviour of various vehicle types," *Transportation Research Procedia*, vol. 25, pp. 4733–4749, 2017. World Conference on Transport Research - WCTR 2016 Shanghai. 10-15 July 2016.

[23] L. Westhofen, C. Neurohr, T. Koopmann, M. Butz, B. Schütt, F. Utesch, B. Neurohr, C. Gutenkunst, and E. Böde, "Criticality metrics for automated driving: A review and suitability analysis of the state of the art," *Archives of Computational Methods in Engineering*, vol. 30, no. 1, pp. 1–35, 2023.

[24] IPG Automotive Group, "IPGRoad InfoFile Description," 2022. Version 11.1.1.