Implementing a dissimilarity metric for scenarios categorization and selection for automated driving systems ⋆

Neeraj Dokania * **Tajinder Singh** ** **Erjen Lefeber** *
Jeroen Ploeg ** **Mohsen Alirezaei** *,**

* *Eindhoven University of Technology*
** *Siemens Digital Industries Software B.V.*

Abstract: To ensure road traffic safety, the safety verification and validation of automated vehicles are of the utmost importance. Scenario-based testing is one of the most popular approaches, as it is cheaper, safer, and faster than on-road testing. The number of possible scenarios encountered by an automated driving system could be virtually infinite due to the complexity and uncertainty of the driving environment. Hence a framework is needed which expresses the degree of dissimilarity between two driving scenarios quantitatively. This work first develops a dissimilarity metric that compares different driving scenarios and secondly, categorizes them to identify the most critical ones in each category. This way, a finite set of non-redundant scenarios are identified which can be used for validating the safety of an automated driving system.

Keywords: Automated driving, Safety of the intended functionality, Verification and validation, Scenario-based testing, Dissimilarity metric, k-medoids clustering

1. INTRODUCTION

Over the past few years, significant effort has been put into developing an automated driving ecosystem of new-age startups, automotive incumbents, and the government. As a result, technology has improved and impressive progress can be seen. However, several technical and ethical challenges limit the wide-scale deployment of automated driving. In 2021, the National Highway Traffic Safety Administration ordered all companies with self-driving vehicles or partially automated systems to report all crashes to the government. The agency found 467 accidents, resulting in 54 injuries and 14 deaths since then. Also in October 2023, the California Department of Motor Vehicles suspended a company's permit for deploying and testing driverless cars, citing 'unreasonable risk to public safety'. These incidents reiterate the importance of safety verification and validation of autonomous vehicles (AVs) before any claims of safety improvement over human drivers can be confidently made.

To demonstrate that the failure rate of AVs in Operational Design Domain (ODD) is statistically significantly lower than the human driver failure rate, autonomous vehicles would require 'billions of miles' of testing [5]. The traditional method of validating the system on roads, while necessary, is incapable of validating the AV's behavior under complex real-life scenarios, thereby demanding safer, cheaper, and faster methods of development. Hence, virtual validation of AVs has gained much traction in the recent past.

ISO (International Organization for Standardization) [1] defines the absence of unreasonable risk resulting from hazardous behaviors related to functional insufficiencies as the safety of the intended functionality (SOTIF). SOTIF classifies scenarios into four areas: Safe-Known (Area 1), Unsafe-Known (Area 2), Unsafe-Unknown (Area 3), and Safe-Unknown (Area 4), primarily aiming to reduce Unsafe-Unknown scenarios. This reduction is done by identifying scenarios in Area 3 and shifting them to Area 2, which is thus covered by verification.

In the literature a scene, a scenario, and relevant terminologies are well-defined [7]. A scene is defined as the evaluation of a concrete scenario at a given time instant. Scenario-based virtual testing has been well explored in the literature. However, this testing approach also cannot practically simulate the 'infinite' number of scenarios encountered by the automated driving systems (ADS). Therefore, attempts have been made to limit this 'infinite' pool of scenarios to a finite set of safety-critical scenarios. These algorithms should not only generate and hence identify scenarios which are 'different' from each other, but they should also select the most critical ones from the set. This raises the questions:

(1) How to select the most critical scenario from a set of generated scenarios?
(2) How to quantitatively define dissimilarity between two driving scenarios?

To address the first question, numerous studies in the literature propose various metrics to quantify how unsafe a sce-

⋆ **This research is conducted at Siemens under the SUNRISE project, funded by the European Union's Horizon Europe Research and Innovation Actions under grant agreement no.101069573.**

nario is. These metrics are either time-based, deceleration-based, or energy-based. Common metrics include Time To Collision (TTC) and its variations, Deceleration Rate to Avoid the Crash (DRAC), and Crash Index (CAI) just to name a few [17]. In contrast, we see a lack of metrics to quantify how unknown a scenario is for the AV. Most of the proposed metrics use a recorded dataset to quantify how unknown a scenario is. Our approach does not consider how unknown the scenario is from the perspective of the AV, but rather how different the scenario is from a limited, previously acquired scenario database. To address this issue, Rajesh et al. [10] and Singh et al. [12] propose an approach to extract the actors' behavior from a real-world dataset and then propose a metric to quantify unknown scenarios.

To address the second question, metrics that can differentiate between scenarios based on a finite set of features are widely explored in the literature. Su et al. [14], Sousa et al. [13], and Tao et al. [15] present surveys on existing measures of similarity in literature. The most commonly used measures from the surveys are Euclidean distance, Lock-step Euclidean distance, Dynamic Time Warping, Longest Common Sub Sequence, Edit distance, Edit Distance with Real Penalty, Normalized Euclidean distance, Fréchet distance, and Hausdorff distance.

The primary emphasis of the approaches by Kerber et al. [6], Bernhard et al. [2], and Pin Nie et al. [9] is on computing dissimilarity based on a scenario trajectory-level formulation. However, Mahadikar et al. [8] proposed a method to compute dissimilarity specifically at the most critical scene, rather than across the entire scenario. This approach is suggested because, from a safety viewpoint, the most relevant information in a scenario is found near the most critical scenes. This reduces the problem domain from trajectory-level analysis to a single scene-level analysis, wherein the most safety-critical scene is identified using a criticality metric. The authors use the minimum distance between actors throughout the scenario as a measure of criticality, as it possibly indicates the risk of an imminent collision. Consequently, the scene with the minimum distance between ego vehicle and other actors is identified as the most safety-critical scene. At the most critical scene of the scenario, the authors choose orientation (potential collision angle, relative heading angle) of the actors for quantifying dissimilarity between the scenarios, and we also follow the same choices.

This work implements a two-step methodology for critical scenario identification, leveraging dissimilarity metrics to categorize and select diverse scenarios. The approach first categorizes scenarios coming out of the optimization algorithm based on discrete features such as actor paths and types. Subsequently, a continuous dissimilarity metric is applied to cluster scenarios within each category, focusing on the most critical scene. This ensures the selection of non-redundant and safety-relevant scenarios. The case study highlights the effectiveness of this method in identifying a diverse and representative set of critical scenarios from an optimization-generated set of scenarios to find novel scenarios.

This paper is organized as follows. Section 2 outlines the workflow and explains different steps including scenario generation, computation of continuous features, calculation of dissimilarity metric based on continuous features, categorization based on discrete features, and selection of the most critical scenario. Section 3 applies the proposed methodology to a set of synthetically generated scenarios. Section 4 summarizes the study findings and post-processes the identified critical scenarios. Finally, Section 5 summarizes the work and provides an outlook on future research.

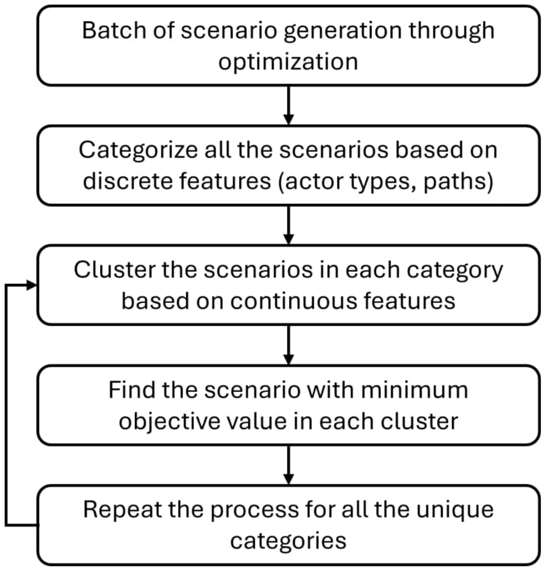

Fig. 1. The main steps of the proposed methodology

2. METHODOLOGY

This section outlines the main steps of the proposed methodology, as illustrated in Fig. 1.

First, a batch of scenarios is generated through an optimization study, as explained in Section 2.1. Next, the scenarios are categorized based on discrete features, as described in Section 2.2. The continuous features are then computed in the most critical scene in all scenarios (Section 2.3). These features are then used to calculate the dissimilarity between two scenarios. Our approach to calculating the dissimilarity metric is inspired by the work of Mahadikar et al. [8], however, the formulation presented in this work, detailed in Section 2.4, differs from that proposed by Mahadikar et al. [8]. This dissimilarity metric forms the basis for clustering the scenarios (Section 2.5). Finally, we identify the most critical scenario in each cluster to find a set of unknown-unsafe scenarios.

2.1 Optimization based generation of scenarios

To generate scenarios that are the most unknown-unsafe and are highly probable in the real world, the optimization method explained in [12] is applied. The objective function $f(\eta)$ used here must satisfy the requirements of novelty and criticality. Hence it is formulated as:

$$f(\eta) = G(P_s(\eta)) \times (2 - \epsilon(\eta) + TTC_{min}(\eta)) \qquad (1)$$

where we assume that a scenario is characterized by the scenario parameters η like actor velocities and positions.ϵ is a Siemens proprietary key performance indicator which

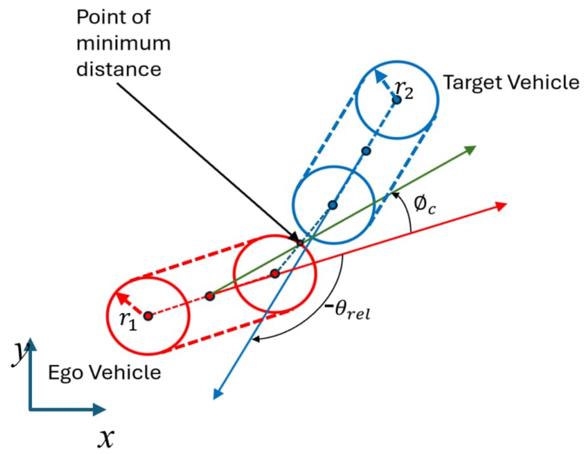

Fig. 2. Schematic view of the ego and target vehicles at the instant of minimum distance along with relevant angles. Counter-clockwise measurement is considered positive. Note that the value of the angles are exemplary.

denotes the unexpectedness of a scenario and TTC_{min} is the minimum time-to-collision, being a measure of how unsafe a scenario is. $G(P_s(\eta))$ is a proprietary function of $P_s(\eta)$ where $P_s(\eta)$ is the probability of occurrence of a scenario. $G(P_s(\eta))$ has an inverse relationship, which means that it gives lower values for higher values of $P_s(\eta)$. It should be noted that ϵ and TTC_{min} are functions of the scenario parameters η, and are evaluated as outputs of the simulation of the scenarios.

2.2 Categorization based on discrete features

The scenarios are first categorized based on discrete features. These features are related to the paths and types of target actor and other relevant actors in a scenario. Relevant actors include actors that influenced the collision, for example, an actor that occluded the collided target from the ego vehicle. Scenarios with different target paths and types are considered entirely distinct. For any two given scenarios with a fixed set of ego and target actor types and paths, if the type or paths of the other relevant actors are different in these scenarios, then they are also considered to be completely different. Furthermore, if there are scenarios with similar discrete features, they could be different in terms of angles as shown in [8], which means they would still be different scenarios, motivating the need of a dissimilarity metric based on the orientation (angles) of the actors.

2.3 Definition and calculation of continuous features

At the most critical scene between ego and target vehicles the continuous features are calculated, as represented in Fig. 2. Both ego and target vehicles are modelled as two circles positioned on their front and rear axles, with diameters equal to the width of the respective vehicles. The distance between the centres of these circles is such that they just fit within the vehicle boundaries. Consequently, the line connecting the centres of the circles represents the vehicle's heading. Note that articulated vehicles are not considered in our work.

The minimum distance between vehicles in a scene is calculated by $\min_{i,j\in\{1,2\}}(d_{ij} - (r_1 + r_2))$. The distances d_{ij} represent the distance between the i^{th} and j^{th} centres of the circles of the two vehicles, where i belongs to the ego vehicle and j belongs to the target vehicle. The radii of the vehicle circles are denoted by r_1 and r_2, as shown in Fig. 2.

The point of minimum distance (PMD) is at the intersection between the circle from the ego vehicle and the line connecting the centres of the representative circles of the two vehicles which are geometrically nearest to each other at this instant. In the PMD, two angles are calculated: the relative heading angle θ_{rel} and the collision/near-miss angle ϕ_c in the frame of the ego vehicle. The physical representation of these angles in a critical scene is illustrated in Fig. 2. The relative heading angle θ_{rel} indicates the orientation difference between two vehicles, showing how they approach each other. The collision/near-miss angle ϕ_c represents the location on the ego-vehicle where a collision or near-miss might occur, such as the side, front, or rear. For each scenario, θ_{rel} is calculated as the angle between the heading vectors of the two vehicles at the minimum distance. The angle ϕ_c is calculated as the angle between the heading of the ego vehicle and the vector from the centre of the ego vehicle to the point of minimum distance (PMD). Since these angles can have an impact on the criticality of the scenario, they are proposed to be used to calculate the dissimilarity metric, for instance, colliding with a target vehicle with front vs back vs side of the ego vehicle will have different influence on the resulting impact. Note that this approach to model the actors can be extended to other types of actors, such as cyclists and pedestrians. For instance, a pedestrian can be represented by a small-radius circle, while a cyclist can be modelled using two small-radius circles positioned at a wheelbase length apart.

2.4 Definition of dissimilarity metric

The dissimilarity metric introduced by Mahadikar et al. [8] integrates both discrete and continuous features into a single measure. In [8], the mesh grid was used as a discrete feature, whereas in our work, we use paths, the type of target actor, and other relevant actors as discrete features. Additionally, we first employ discrete features for categorization and then apply continuous features to refine categorization within each category. The method used here is represented in Fig.3.

Consider any two scenarios $S1$ and $S2$ with the following values of continuous features;

$$S1 : (\theta_{rel1}, \phi_{c1}) \text{ and } S2 : (\theta_{rel2}, \phi_{c2}).$$

First, the cosine dissimilarity [3] is calculated according to

$$\Delta H = [1 - \cos(\theta_{rel1} - \theta_{rel2})]/2 \quad (2)$$

$$\Delta C = [1 - \cos(\phi_{c1} - \phi_{c2})]/2 \quad (3)$$

Here the cosine similarity index is halved, because it returns values in the range [0,2] but it is desired to be reduced to [0,1] for simplicity and intuitive understanding of dissimilarity. Then the dissimilarity metric $\delta(S1, S2)$ is calculated as;

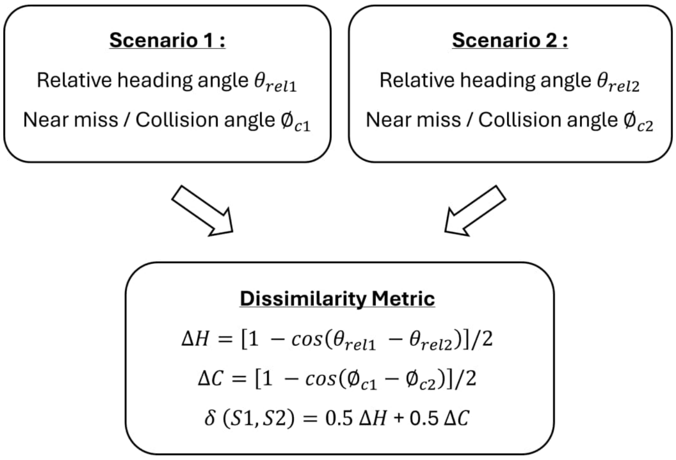

Fig. 3. Framework for continuous dissimilarity metric calculation for two scenarios in a category

$$\delta(S1, S2) = w_1 \Delta H + w_2 \Delta C \quad (4)$$

$$w_1 + w_2 = 1 \quad (5)$$

The weights (w_1, w_2) of both the angles can change based on the environment being analyzed. For instance, in the case of highway driving, the difference in heading angles of the scenarios would not be too large, hence it should have a lower weighting. However, in our case of an intersection (Fig. 5), equal weightings ($w_1 = w_2 = 0.5$) have to be given to both the angles.

$\delta(S1, S2)$ closer to 1 shows a higher level of dissimilarity between the scenarios whereas the values closer to 0 indicate otherwise.

2.5 Clustering based on continuous features

To cluster the scenarios with the formulated dissimilarity metric, k-medoids clustering is applied [4]. This is an unsupervised learning algorithm used for data clustering, which groups unlabeled data points into clusters, even with a custom distance metric. This type of clustering requires a priori choosing the number of clusters k. The clustering process is conducted iteratively for various values of k, and the cluster quality is evaluated by calculating the silhouette metric for each iteration. The silhouette metric is a measure of how compact and isolated a cluster is from other clusters [11]. The value of k which gives the highest value of the silhouette metric is therefore chosen. An advantage of the silhouette metric is that it can be applied to any distance metric, making it easy to implement with the dissimilarity metric. Hence the best value of k, based on the silhouette score, is then used to cluster the set of scenarios.

Fig. 4 gives a visual representation of what the proposed methodology aims to achieve.

3. IMPLEMENTATION

In this section, the proposed methodology is applied to a case study of an intersection. The intersection is presented in Fig. 5.

To apply the proposed methodology to a case study, a synthetic set of scenarios is generated by the method proposed by [12] as shown in Fig. 6. Firstly, data collection of trajectories is performed through instrumented vehicles as well as data-recording setups like static drone footages or infrastructure-mounted sensors. This data is then processed to remove sensor noise and extract full trajectories.

Next, actors' behavior extraction is performed from the recorded trajectory data in each scene. This is done to primarily identify a relevant and finite set of features which can explain the behavior of the actors. This is done by describing the road layout as a graph, where the nodes of the graph are defined by unique entry, exit and intermediate points in the defined scene. Then an average path of the actors is made based on the probability distribution of the parameters. The trajectories for all the non-ego actors are now parameterized and scenarios are generated by varying the parameters that affect the scenarios. These parameters include but are not limited to entrance times, velocity at different sections and lateral offset of the vehicles [12].

The scenario is modelled in Simcenter Prescan [1], a physics-based simulation software for the virtual validation of autonomous vehicles. The Prescan Graphical User Interface is used to model the actors described above, along with the road layout. Scenario parameters are varied using a script that uses the data model API feature of Prescan. Simcenter HEEDS design exploration and optimization software [2] offers the SHERPA algorithm [16] which solves the optimization problem to find unknown-unsafe scenarios using Eq. 1. It also automatically optimizes the hyperparameter values of the individual optimization algorithms. SHERPA can handle mixed discrete-continuous spaces and nonlinear, non-convex fitness functions. Consequently, the Siemens Prescan HEEDS tool-chain was utilized to run simulations and optimize the scenario parameters. Multiple HEEDS studies were conducted with varying paths for all actors in the scenarios. Each HEEDS study focuses on finding the most critical scenarios for a different ego-main target combination.

4. RESULTS AND DISCUSSION

This section presents the results from a single HEEDS study where the paths of both the ego vehicle and the main target vehicle are fixed. In the previously described use case, HEEDS generated a total of 1,000 scenarios, of which 395 were found to meet the constraints, e.g., collision of non-ego actors or lack of interaction between ego and target vehicles.

For the 395 feasible scenarios, 12 categories were identified based on discrete features (type and path of non-target actors). The scenarios within each category were then clustered based on a dissimilarity metric. The most critical scenario, i.e., the scenario with the minimum objective score $f(\eta)$, was identified in each cluster within a category. For example, in a particular category of scenarios, three distinct scenarios were found and reported based on the dissimilarity metric, as shown in Fig. 7. This approach

[1] Simcenter Prescan software - https://plm.sw.siemens.com/en-US/simcenter/autonomous-vehicle-solutions/prescan/
[2] Simcenter HEEDS - https://plm.sw.siemens.com/en-US/simcenter/integration-solutions/heeds/

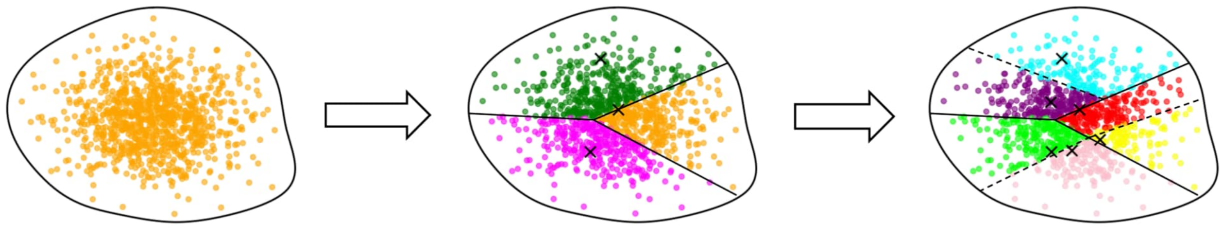

Fig. 4. The entire set of scenarios is categorized on two levels. First, the set of discrete features categorizes the scenarios into three categories (solid lines). Each category is then further clustered (dotted lines) based on a dissimilarity metric using continuous features. This process identifies six unique, critical scenarios, marked by a black cross. Note that the numbers three and six are illustrative.

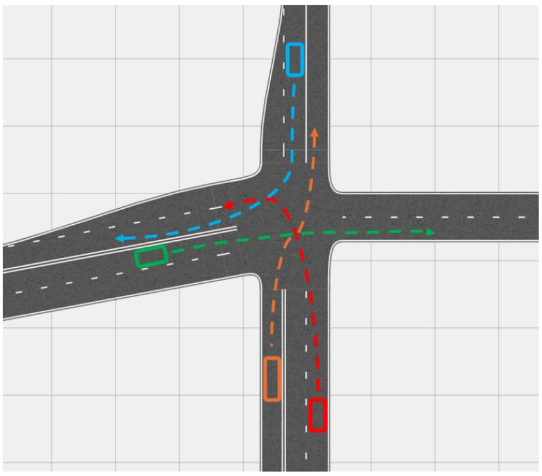

Fig. 5. Top-view of the simulated intersection with exemplary actors' paths. The ego vehicle is colored red, the target vehicle is blue and there are two other actors shown in orange and green. The dotted lines show their respective paths.

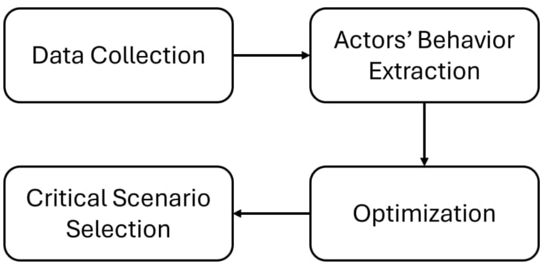

Fig. 6. The main steps of scenario generation process

results in higher number of scenarios within each category compared to using only discrete features.

A total of 28 distinct safety-critical scenarios are found through this method. This is a significant increase from what we obtained when we only used discrete features for categorization, i.e., 12 scenarios. Fig. 8 shows the snapshots of the identified scenarios in a particular category. It can be seen from the snapshots that these identified scenarios are dissimilar to each other based on how the ego vehicle interacts with the other actors in the most critical scene. For instance, although the heading angles of the ego and target vehicles are comparable in Scenarios A and B, they collide with each other in very different ways, as reflected in their collision angles, which are shown in the snapshots. Likewise, the collision/near-miss angles

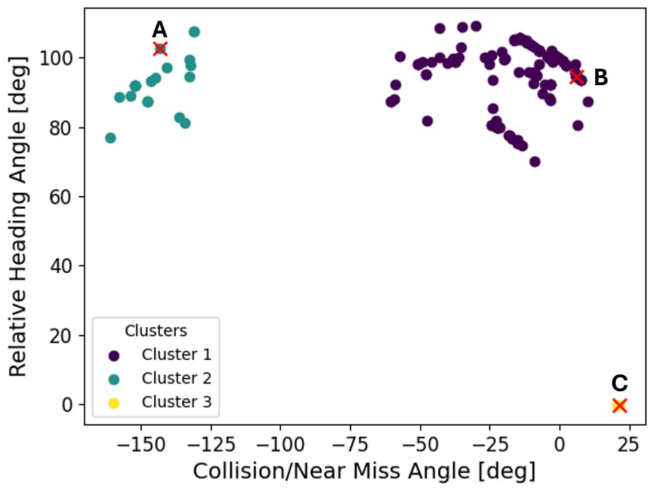

Fig. 7. Clustering plot with identified critical scenarios in a particular category marked with a red cross in each cluster

are comparable in Scenarios B and C, but their relative orientation in the most critical scene is quite different, thereby suggesting dissimilarity between these scenarios.

5. CONCLUSION

This paper presents a methodology and case study for identifying diverse critical scenarios using dissimilarity metrics. By employing a two-level categorization, first based on discrete features and then continuous features, a higher number of critical scenarios is identified, compared to using discrete features only. The case study demonstrates the efficacy of this approach in deriving a set of diverse scenarios that capture key safety challenges in the ODD.

Currently, the heading and collision/near-miss angles used in the dissimilarity metric implementation are based on the approximations of actor bounding boxes. These approximations can be further refined in future work by integrating detailed bounding box data from Prescan simulations, leveraging object-level sensor models. Additionally, extending the evaluation to consider events occurring before the collision or near-miss would allow for temporal analysis of scenario criticality. Further enhancements could include adapting the methodology to different ODDs, such as highways or urban environments, to generalize its application.

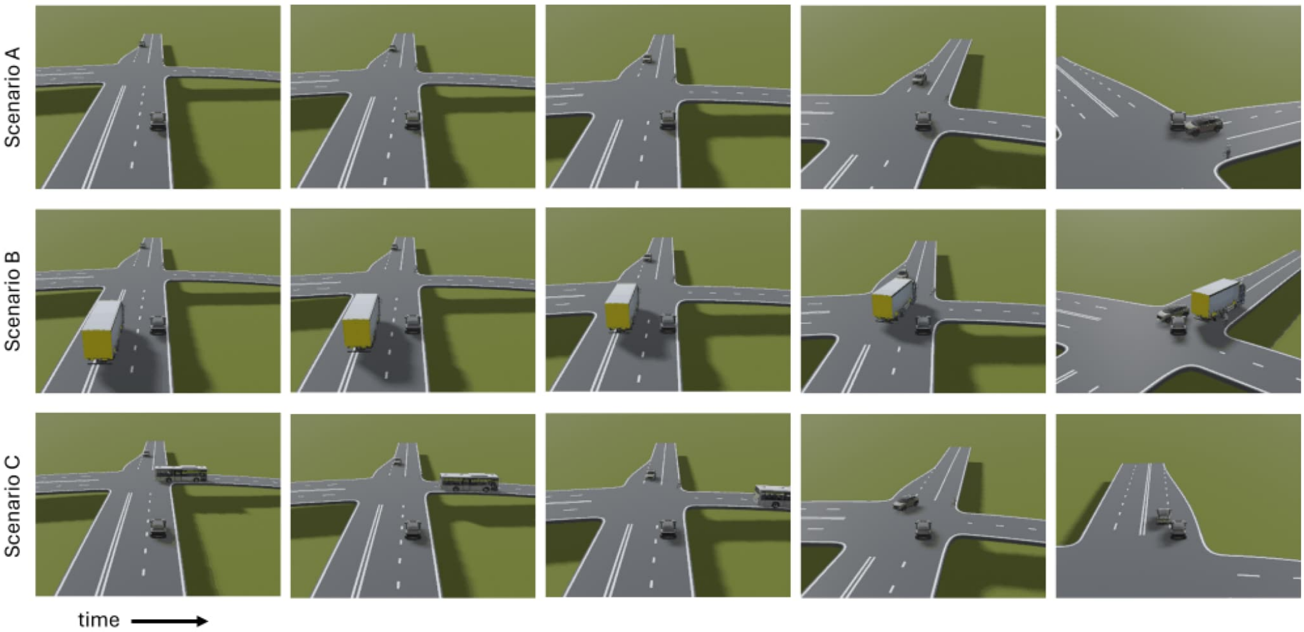

Fig. 8. Examples of identified critical scenarios. Animation snapshots of Designs A, B, and C respectively.

REFERENCES

[1] Iso 21448: Road vehicles — safety of the intended functionality.

[2] J. Bernhard, M. Schutera, and E. Sax. Optimizing test-set diversity: Trajectory clustering for scenario-based testing of automated driving systems. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1371–1378, IN, USA.

[3] J. Han, M. Kamber, and J. Pei. 2 - getting to know your data. In Jiawei Han, Micheline Kamber, and Jian Pei, editors, *Data Mining (Third Edition)*, The Morgan Kaufmann Series in Data Management Systems, pages 39–82. Morgan Kaufmann, Boston, third edition edition, 2012.

[4] X. Jin and J. Han. *K-Medoids Clustering*, pages 564–565. Springer US, Boston, MA, 2010.

[5] 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*, 94:182–193, 2016.

[6] J. Kerber, S. Wagner, K. Groh, D. Notz, T. Kühbeck, D. Watzenig, and A. Knoll. Clustering of the scenario space for the assessment of automated driving. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 578–583, Las Vegas, NV, USA.

[7] K. T. Kurian, N. Rajesh, E. Lefeber, J. Ploeg, N. van de Wouw, I. Besselink, and M. Alirezaei. Formalizing vocabulary for scenario-based testing of automated driving systems: From semantics to mathematics. In *27th IEEE International Conference on Intelligent Transportation Systems (ITSC)., 2024*.

[8] B. B. Mahadikar, N. Rajesh, K. T. Kurian, E. Lefeber, J. Ploeg, N. van de Wouw, and M. Alirezaei. Formulating a dissimilarity metric for comparison of driving scenarios for automated driving systems. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 1091–1098, Jeju Island, Korea, Republic of.

[9] P. Nie, Z. Chen, N. Xia, Q. Huang, and F. Li. Trajectory similarity analysis with the weight of direction and k-neighborhood for ais data. *ISPRS International Journal of Geo-Information*, 10(11), 2021.

[10] N. Rajesh, E. van Hassel, and M. Alirezaei. Unknown-unsafe scenario generation for verification and validation of automated vehicles. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1759–1764, Bilbao, Spain.

[11] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[12] T. Singh, E. van Hassel, A. Sheorey, and M. Alirezaei. A systematic approach for creation of sotif's unknown unsafe scenarios: An optimization based method. *SAE Technical Paper Series*, 2024.

[13] R. S. De Sousa, A. Boukerche, and A. A. F. Loureiro. Vehicle trajectory similarity: Models, methods, and applications. *ACM Comput. Surv.*, 53(5), September 2020.

[14] H. Su, S. Liu, B. Zheng, X. Zhou, and K. Zheng. A survey of trajectory distance measures and performance evaluation. *The VLDB Journal*, 29(1):3–32, October 2019.

[15] Y. Tao, A. Both, R. I. Silveira, K. Buchin, S. Sijben, R. S. Purves, P. Laube, D. Peng, K. Toohey, and M. Duckham. A comparative analysis of trajectory similarity measures. *GIScience & Remote Sensing*, 58(5):643–669, 2021.

[16] R. C. Technology. Sherpa – an efficient and robust optimization/search algorithm. 2017.

[17] C. Wang, Y. Xie, H. Huang, and P. Liu. A review of surrogate safety measures and their applications in connected and automated vehicles safety modeling. *Accident Analysis Prevention*, 157:106157, 2021.