

Diverse Sequential Metamorphic Testing of Autonomous Driving: A Scenario-Based Empirical Study

Huai Liu
Swinburne University of Technology
Melbourne, Australia
hliu@swin.edu.au

Quang-Hung Luu
Swinburne University of Technology
Melbourne, Australia
hluu@ieee.org

Caslon Chua
Swinburne University of Technology
Melbourne, Australia
cchua@swin.edu.au

T. H. Tse
The University of Hong Kong
Pokfulam, Hong Kong
thtse@cs.hku.hk

Tsong Yueh Chen
Swinburne University of Technology
Melbourne, Australia
tychen@swin.edu.au

Abstract—Ensuring the safe decisions of autonomous vehicles (AVs) is of utmost importance, given their increasing popularity in public roads today. Sequential metamorphic testing has been successfully applied to reveal failures in various AV systems by verifying sequences or groups of metamorphic relations (MRs) among multiple of inputs and outputs. Having said that, new MRs are still required to address the complexity of driving scenarios. In this paper, we present diverse sequences of MRs for testing perception systems in AVs. Systematic applications of these MRs allow us to detect more failures from various deep-learning-based AV systems in response to diverse road objects, environment factors, and their combinations. The new MRs also help us gain more insights into these systems. The success of our study demonstrates that the journey to identify new MRs for testing autonomous driving is challenging yet significant and rewarding.

Index Terms—Metamorphic Testing, Metamorphic Relations, Autonomous Vehicles

I. INTRODUCTION

Millions of autonomous vehicles (AVs) from partial automation to full self-driving capability have been operating on public roads today. Ensuring their safe decisions is, at all time, of utmost importance to road users. However, determining the correctness of a decision made by an AV is never an easy task, if not infeasible, due to the fact that it encounters an endless variety of real-world scenarios [1, 2]. From software testing perspective, AVs are untestable systems where it is infeasible to evaluate their outputs given any possible inputs, which is normally referred to as the oracle problem in the context of software testing. Addressing this challenge remains a fundamental issue in autonomous driving [3, 4].

Metamorphic testing (MT) has emerged as an effective approach to tackle this challenge [1, 2, 5, 6]. Instead of determining the expected outputs corresponding to individual

inputs for the system under test in traditional testing, it makes use of relations between multiple inputs and outputs of system under test to determine the correctness of decisions. Such relations, called metamorphic relations (MRs), are identified from the properties of system under test to help detect system failures [7, 8]. Zhou and Sun [5] applied MT in combination with fuzzing to Baidu’s Apollo self-driving car where test cases of modified LiDAR input scenarios are evaluated against the original unmodified test cases in measuring Apollo’s perception. They discovered previously unknown bugs in the object detection of Apollo and reported them eight days before the fatal accident of a Uber self-driving car that killed a woman crossing the road. DeepTest [1], on the other hand, was one of the first tools to automatically generate new test inputs for testing deep-learning-based driving models by adopting transformations that mimic the environmental changes including weather, lighting conditions, and object occlusions. It helped reveal thousands of errors in the steering decisions, which could lead to deadly crashes. Another study [9] made use of a model based on generative adversarial networks (GAN) to generate images of the road under diverse weather conditions such as rain, snow or fog to evaluate the consistency of driving decisions. The GAN-based approach revealed thousands of inconsistencies in models under test. These studies underpinned the importance and great promises of MT in testing perception systems for autonomous driving.

More recently, researchers [6, 10, 11] have further developed more structured and systematic frameworks for MT of autonomous vehicles. Luu *et al.* [6] proposed the SMART framework to construct MRs in sequences or groups so that outputs from one test can be utilized in another MRs, enabling a deeper and more efficient exploration of the AV’s failures. The effectiveness of SMART was further demonstrated on Autoware, a well-known fully autonomous driving stack, where sequential metamorphic tests uncovered hazardous flaws in

maintaining safe distance. Zhang *et al.* [10] have developed scenario-driven MRs for testing AVs. They used MR patterns of scenario template alongside an iterative loop that refines test scenarios and MRs based on found issues. Recently, Yang *et al.* [11] introduced MT-Nod to test the optimal decisions in interactive planning scenarios. Another emerging approach for MT is to make use of large language models to develop MRs for the testing of various systems, including AVs [12, 13].

Despite the great success of MT in testing AVs, a huge technical gap still exists, that is, the current available MRs are not sufficient to cover the diverse and complicated driving scenarios. In this study, we develop several rich sets of new MRs. More specifically, 75 new MRs have been developed based on 18 patterns, mainly aimed at capturing driving scenarios related to various positions of different objects and/or distinct environmental conditions. Our objectives are twofold. On the one hand, it is demonstrated that our new MRs are effective in revealing new failures of systems under test. On the other hand, we make use of these MRs to gain new insights into these systems under test.

The rest of our paper is organized as follows: In Section II, we present the methodology used in this study, which consists of an introduction of the background knowledge of MT, the new MRs we develop and use for this study, and the design and settings for the experiments. The detailed experimental results are given in Section III, where we also summarize the main findings, including the failure-detection effectiveness of new MRs and the new insights gained for the systems under test. The paper is enclosed by the concluding remarks in Section IV.

II. METHODOLOGY AND EXPERIMENTS

A. Background on metamorphic testing

Metamorphic relations (MRs) are relations among multiple inputs and outputs of system under test (SUT). They are developed from necessary properties of the targeted algorithm to be implemented for the SUT. In other words, the system, if implemented correctly, must uphold these MRs. Otherwise, if an MR is violated, a failure is considered to be revealed from the system. MT differs from the traditional techniques in the sense that instead of requiring knowledge of the correct output for an individual test case, MT checks whether the outputs of the source (original) test cases and the follow-up test cases (generated by transforming source test cases based on MRs) satisfy these relations.

To illustrate the basic process of MT, consider the example of a program \hat{g} that implements an algorithm g finding the shortest distance between node A and node B in the graph G . It is very tedious to verify the correctness of \hat{g} on an undirected graph G of N ($N \geq 3$) nodes, as there are $[e \cdot (N - 2)!]$ possible paths, assuming all nodes are connected. With MT, we can make use of an MR $g(A, C) + g(C, B) \geq g(A, B)$, where C is another node different from A and B , to test the correctness of SUT. If the relation among outputs of the program does not hold, that is, $\hat{g}(A, C) + \hat{g}(C, B) < \hat{g}(A, B)$, we know that \hat{g} is faulty. In other words, if \hat{g} is correct, it must

hold the relation $g(A, C) + g(C, B) \geq g(A, B)$ for arbitrary values of A, B , and C . Note that the inputs to \hat{g} are not only target nodes A, B but also the graph G , and there could be multiple shortest paths between A and B in the same graph.

The process of testing a program \hat{g} with MT can be summarized as the following steps:

- 1) Identify the MR from the necessary properties of target algorithm g . In the above example, it is $g(A, C) + g(C, B) \geq g(A, B)$ for any values of A, B, C and any graph G .
- 2) Prepare the inputs to the source test cases. Here, it is the specific nodes A, B and the specific graph G .
- 3) Execute the program \hat{g} to obtain source outputs, which is the distance $\hat{g}(A, B)$.
- 4) Use these source inputs and, if necessary, source outputs to construct the follow-up inputs according to the MR. Herein, we need two follow-up test cases, which consist of nodes A, C and nodes C, B , respectively.
- 5) Execute the program to obtain follow-up outputs. In the above example, we have two follow-up outputs $\hat{g}(A, C)$ and $\hat{g}(C, B)$.
- 6) Compare these multiple inputs and against the MR by replacing g by \hat{g} . If the relation $\hat{g}(A, C) + \hat{g}(C, B) \geq \hat{g}(A, B)$ is not satisfied, the program \hat{g} is determined to be faulty, that is, a failure is revealed by this MR.

B. Motivation and research questions

Previous studies on MT for autonomous driving [1, 2, 5, 6] normally made use of the MRs that are based on the first level of information from the SUT. They established how the SUTs respond to hazardous scenarios that involve new objects, changing environmental scenes and their combinations. Table I presents a summary of MRs used in previous studies [1, 2], which can be grouped into patterns, namely Metamorphic Relation Input Patterns (MRP), an abstraction that characterizes a set of similar MRs [14].

Having said that, these MRs do not consider the second layer of helpful information that can be used for testing. For example, traditional MR may show that changing the color of the leading car can be used to examine the correctness and robustness of the SUT. However, the SUT may respond differently to the leading car in a red color instead of a blue one [6]. Hence, it is necessary to develop a new set of MRs to harness deeper layer of information ignored in conventional MRs.

In this study, we identify new MRs that can help extract new information from the SUTs for self-driving cars. They are described in Section II-C. Specifically, we aim to address the following two research questions:

- Are these new MRs effective in failure detection?
- Can MRs give us more useful information about the SUT?

C. New metamorphic relations

Our MRs are identified with the intuition that deeper insights from the SUT can be obtained by considering them

TABLE I
EXISTING MRS THAT HAVE BEEN USED IN PREVIOUS STUDIES [1, 2, 6] WHERE SOURCE TEST CASE IS AN IMAGE OF DRIVING SCENE.

MRPs	Follow-up test case	Condition	Note
Object-based scenarios			
eMRP _{O1}	Add a forward-moving car closely in front of the AV	$M_C(g) < \theta_e$	Change SA to minimize collision
eMRP _{O2}	Add an approaching car closely in front of the AV	$M_C(g) < \theta_e$	Change SA to minimize collision
eMRP _{O3}	Add a car moving in a direction closely in front of the AV	$M_C(g) < \theta_e$	Change SA to minimize collision
eMRP _{O4}	Add a forward-moving car with a color in front of the AV	$M_C(g) < \theta_e$	Change SA to minimize collision
eMRP _{O5}	Add a forward-moving vehicle in front of the AV	$M_C(g) < \theta_e$	Change SA to minimize collision
eMRP _{O6}	Add an object in front of the AV	$M_C(g) < \theta_e$	Change SA to minimize collision
Environment-based scenarios			
eMRP _{E1}	Transform the scene to a rainy weather	$M_U(g) < \theta_e$	Robust against environmental change
eMRP _{E2}	Transform the scene to a gravel road	$M_U(g) < \theta_e$	Robust against environmental change
eMRP _{E3}	Transform the scene to a snowy weather	$M_U(g) < \theta_e$	Robust against environmental change
eMRP _{E4}	Transform the scene to a darker lightning condition	$M_U(g) < \theta_e$	Robust against environmental change
eMRP _{E5}	Transform the scene to a brighter lightning condition	$M_U(g) < \theta_e$	Robust against environmental change
eMRP _{EX}	Transform the scene to a rainy weather with gravel road	$M_U(g) < \theta_e$	Robust against environmental change
Combined (object- and environment-based) scenarios			
eMRP _{C1}	Add a car and transform to a rainy weather	$M_C(g) < \theta_e$	Change SA to minimize collision
eMRP _{C2}	Add a car and transform to a gravel road	$M_C(g) < \theta_e$	Change SA to minimize collision
eMRP _{C3}	Add a car and transform to a snowy weather	$M_C(g) < \theta_e$	Change SA to minimize collision
eMRP _{C4}	Add a car and transform to a darker lightning condition	$M_C(g) < \theta_e$	Change SA to minimize collision
eMRP _{C5}	Add a car and transform to a brighter lightning condition	$M_C(g) < \theta_e$	Change SA to minimize collision
eMRP _{CX}	Add a car and transform to a rainy weather with gravel road	$M_C(g) < \theta_e$	Change SA to minimize collision

TABLE II
NEW MRS IDENTIFIED FROM OUR APPROACH.

MRPs	(Source, follow-up) test cases	Condition	Note	Total
Object-based scenarios				
nMRP _{O1}	(center, left)	$M_L(g) < \theta_n$	Steer to left-side of the lead car already on opposite lane	4 positions
	(center, right)	$M_U(g) < \theta_n$	Steer to right-side of the lead car moving toward side road	4 positions
nMRP _{O2}	(center, left)	$M_L(g) < \theta_n$	Steer to left-side of the facing car already on opposite lane	4 positions
	(center, right)	$M_U(g) < \theta_n$	Steer to right-side of the facing car moving toward side road	4 positions
nMRP _{O3}	(forward, anticlockwise:045)	$M_L(g) < \theta_n$	Steer to left-side of the lead car moving to opposite lane	7 angles
	(forward, anticlockwise:090)	$M_L(g) < \theta_n$	Steer to left-side of the car moving across to opposite lane	
	(forward, anticlockwise:135)	$M_L(g) < \theta_n$	Steer to left-side of the facing car moving to opposite lane	
	(forward, facing)	$M_C(g) < \theta_n$	Do not steer to avoid crash with the facing car	
	(forward, clockwise:135)	$M_R(g) < \theta_n$	Steer to right-side of the facing car moving to side road	
	(forward, clockwise:090)	$M_R(g) < \theta_n$	Steer to right-side of the car moving across to side road	
	(forward, clockwise:135)	$M_R(g) < \theta_n$	Steer to right-side of the lead car moving to roadside	
nMRP _{O4}	(red, color)	$M_C(g) < \theta_n$	Steer inconsistently for the lead car in a different color	2 colors
nMRP _{O5}	(car, vehicle)	$M_C(g) < \theta_n$	Steer inconsistently for the similar vehicle	2 cars
nMRP _{OX}	(dog, kangaroo)	$M_C(g) < \theta_n$	Steer inconsistently for the similar object	1 pair
Environment-based scenarios				
nMRP _{E1}	(rain:light, rain:level)	$M_C(g) < \theta_n$	Steer inconsistently for diverse rain conditions	2 levels
nMRP _{E2}	(gravel:0.2, gravel:level)	$M_C(g) < \theta_n$	Steer inconsistently for diverse road conditions	4 levels
nMRP _{E3}	(snow:0.2, snow:level)	$M_C(g) < \theta_n$	Steer inconsistently for diverse snow conditions	4 levels
nMRP _{E4}	(brightness:0.2, brightness:level)	$M_C(g) < \theta_n$	Steer inconsistently for diverse brightness conditions	4 levels
nMRP _{E5}	(darkness:0.2, darkness:level)	$M_C(g) < \theta_n$	Steer inconsistently for diverse darkness conditions	4 levels
nMRP _{EX}	(gravel:1+rain:none, gravel:1+rain:level)	$M_C(g) < \theta_n$	Steer inconsistently for combined road and rain conditions	3 level
Combined (object- and environment-based) scenarios				
nMRP _{C1}	(car+rain:none, car+rain:level)	$M_C(g) < \theta_n$	Steer inconsistently for diverse rain conditions	3 levels
nMRP _{C2}	(car+gravel:none, car+gravel:level)	$M_C(g) < \theta_n$	Steer inconsistently for diverse road conditions	5 levels
nMRP _{C3}	(car+snow:none, car+snow:level)	$M_C(g) < \theta_n$	Steer inconsistently for diverse snow conditions	5 levels
nMRP _{C4}	(car+brightness:none, car+brightness:level)	$M_C(g) < \theta_n$	Steer inconsistently for diverse brightness conditions	5 levels
nMRP _{C5}	(car+darkness:none, car+darkness:level)	$M_C(g) < \theta_n$	Steer inconsistently for diverse darkness conditions	5 levels
nMRP _{CX}	(car, car+gravel:1+rain:level)	$M_C(g) < \theta_n$	Steer inconsistently for combined rain and gravel conditions	3 levels

in a sequential way [6]. Having said that, the first study of the SMART framework [6] presents a way to do it, but the MRs used in that study did not fully cover the diverse and complicated driving scenarios.

We are thus motivated to develop new MRs for testing deep learning models for autonomous driving perceptions under various scenarios. MRs are grouped into 18 patterns (MRPs), and the MRPs are further classified into the three categories associated with the objects, the environment, and their combinations as shown in Table II.

- In the object-based category of scenarios, there were six groups, each of which contains patterns related to the validation of SUTs against the response to objects, including diverse horizontal positions of leading car moving forward (nMRP_{O1}), diverse horizontal positions of approaching car (nMRP_{O2}), diverse moving directions of the car in front (nMRP_{O3}), diverse colors of leading car (nMRP_{O4}), diverse types of leading vehicle (nMRP_{O5}), and diverse types of objects detected in the road (nMRP_{O6}).
- In the environment-based category of scenarios, the SUTs are tested against the groups associated with diverse intensities of rain (nMRP_{E1}), diverse levels of severity of gravel (nMRP_{E2}), diverse intensities of snow (nMRP_{E3}), diverse levels of brightness (nMRP_{E4}), diverse levels of darkness (nMRP_{E5}), and the combined effects of road and rain conditions (nMRP_{EX}).
- In the combined (object and environment-based) scenarios, the test cases were setup by injecting a leading car into scene followed by applying an environmental effect. Hence, in this category, the SUTs were verified against the groups of scenarios associated a source test case of base intensity/level and follow-up ones with the leading car under diverse intensities of the rain (nMRP_{C1}), the severity of gravel (nMRP_{C2}), diverse intensities of the snow (nMRP_{C3}), diverse levels of brightness (nMRP_{C4}), diverse levels of darkness (nMRP_{C5}), and the combined effects of gravel and rain (nMRP_{CX}).

In total, we have 75 individual MRs, grouped in 18 MPRs. Similar to DeepTest [1] and SMART [6], the violation of each MR requires a measure $M(g)$ to quantify the violation. In particular, we adopt four measures, which are defined as follows:

- The *unchanged* measure ($M_U(g)$) is to quantify a small deviation of output (i.e. steering angle), that is,

$$M_U(g) = \frac{1}{2}|\hat{\delta}(g)| \quad (1)$$

- The *change* measure ($M_C(g)$) is to quantify a large deviation, that is,

$$M_C(g) = \begin{cases} \gamma - \frac{1}{2}|\hat{\delta}(g)| & \text{if } \frac{1}{2}|\hat{\delta}(g)| < \gamma \\ \gamma & \text{otherwise} \end{cases} \quad (2)$$

- The *rightward change* measure ($M_R(g)$) is to quantify a rightward deviation, that is,

$$M_R(g) = \begin{cases} \frac{1}{2}|\hat{\delta}(g)| & \text{if } \hat{\delta}(g) < 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

- The *leftward change* measure ($M_L(g)$) is to quantify a leftward deviation, that is,

$$M_L(g) = \begin{cases} \frac{1}{2}|\hat{\delta}(g)| & \text{if } \hat{\delta}(g) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where γ is an adjustable parameter for the maximum of changes allowed, set by the framework users.

We adopt the Heaviside step function to determine the violation of MR, that is,

$$U(g) = \begin{cases} 1, & M(g) > \theta \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where 1 is to assert that the MR is violated, and 0 is to indicate otherwise or inclusiveness. Similarly, we use threshold θ_n to account for the violation of MR.

To evaluate the overall failure-detection effectiveness (FDE) of an MR, we adopt a commonly used measure in MT known as the *ratio of violations* [7], or simply r_{FDE} . It calculates the ratio of metamorphic groups of source and follow-up test cases that detect failures (n_V) to the total number of groups of test cases adopted (n_{MG}), that is:

$$r_{FDE} = \frac{n_V}{n_{MG}} \quad (6)$$

D. Experimental setup

In this study, we selected three deep learning models that steer self-driving cars, namely Chauffeur, Rambo, and Autumn¹, as SUTs for the experiments. Being the winners of Udacity Self-Driving Car Challenge 2, they have been used extensively in other studies [1, 2, 6]. The architectures of the Chauffeur and Autumn models are essentially based on the combinations of convolutional neural networks (CNN) and recurrent neural networks (RNN) for extracting dominant features and learning patterns. Meanwhile, the Rambo model is basically an integration of three different CNNs. In testing these systems, we used the Udacity Test Challenge dataset. It consists of 5615 images recorded with a resolution of 640 × 480 captured by the front camera of the car. We generated new test cases from each single input image. We also inspected all individual images to assure that they are reasonably realistic. We conducted our experiments with these models on OzSTAR, the supercomputing facility of Swinburne University of Technology. Each compute node was geared by an 18-core Intel processor and two NVIDIA P100 graphic computing units. We configured the system with TensorFlow, Keras, and OpenCV as required by these models. The assessment with Udacity datasets showed that they yielded small mean absolute errors (MSE) and high Pearson correlation coefficients [6],

¹<https://github.com/udacity/self-driving-car>

indicating that the models were well configured and ready to be used for further testing. We used the SMART framework [6] to generate test cases and evaluate MRs.

III. RESULTS

A. Failure-detection effectiveness of new MRs

In this section, we evaluate whether the new MRs are effective in the failure detection. To this end, we have analyzed 75 new MRs in all three categories, namely environment-based, object-based, and combined scenarios. By considering various MRPs and a boarder range of scenarios, our study could give us a more confident answer to this research question.

First, it is worth mentioning that the violation of an MR is subject to the used threshold value. Fig. 1 presents the r_{FDE} of the Chauffeur model for various thresholds. With $\theta_n = 0.02$, the average FDE value was 22.0% of scenarios considered. The large spectrum of the box plot indicates that the failure-detection effectiveness of MRs varied from one to another, depending on both the nature of MRs and metamorphic groups of test cases. The lower the threshold was, the more effective the MRs were. Lowering the threshold ($\theta_n = 0.01$) gave a higher percentage of failure-detection effectiveness (37.8%); and in contrast, leveling up the threshold would reduce the effectiveness (e.g., to 7.3% for threshold $\theta_n = 0.05$). With the highest threshold considered in this study ($\theta_n = 0.2$), the effectiveness was 0.4%. Equivalently, we have revealed a total number of about 2000 ($= 5615 \times 0.4 \times 89\%$) undesirable responses over the total of 5615 input scenarios. The trade-off is that a higher threshold gave us a more reliable estimate, as the difference became large enough. The original study of the SMART framework [6] have demonstrated that $\theta_n = 0.02$ is a fair setting for balancing this trade-off thereby providing a statistically reliable estimation of the testing effectiveness. Thus, this study used the same setting ($\theta_n = 0.02$) as the default threshold for computing r_{FDE} for each MR.

In particular, the FDE of each system would be different for different categories of scenarios. The Autumn model seemed to be robust in all categories of scenarios, with the effectiveness ranging from 11.4% to 14.9% (Fig. 2). In contrast, MRs related to the combined scenarios in the Chauffeur and Rambo models were the most effective ones, with r_{FDE} as high as 30.2% and 28.1%, respectively. In all models, the FDE values for the object-based scenarios were smaller than those for other categories of scenarios (Fig. 2).

The FDE values also varied significantly from MR to MR in each model, as can be seen in Fig. 3. For the Autumn and Rambo models, the largest effectiveness values in the object-based scenarios were observed in the MRs related to forward moving object (whose r_{FDE} values were 73.1% to 90.8%, respectively, shown in Fig. 3). Among the environment-based scenarios, MRs associated with the rainy and completely dark conditions had the highest FDEs. In the combined scenarios, the MRs related to gravels, snows, and rains had higher r_{FDE} values than other MRs in the Chauffeur and Rambo models. In general, the MRs in the combined scenarios were

more effective than the MRs for purely environment-based scenarios.

B. Insights of findings revealed by new MRs

In this section, we discuss the insights of findings revealed by our new MRs that had not been detected by existing MRs or observed earlier in these SUTs.

1) *Insights into responses to varying objects:* New MRs allowed us to understand how the ADS models responded differently to scenarios related to the position of the leading car. When the leading car was on the right (relative to the center of the windscreen of the car under test, a.k.a. the ego car), it tended to change lane or move to the shoulder (breakdown or emergency) lane. When the leading car was on the left, it was moving head-to-head or from the opposite lane toward the ego car. Our new MRs revealed that the Chauffeur model had more issues when the leading car was on its right other than on its left. The average value of r_{FDE} for MGs related to the left was 10.3% ($= (7.8\% + 9.9\% + 11.5\% + 12.1\%) / 4$), which was just about a half of the value on the right of 17.8% ($= (18.1\% + 14.6\% + 16.7\% + 21.6\%) / 4$) (Fig. 3). In contrast, the Autumn model had more problems on the left (16.2% ($= (9.4\% + 17.3\% + 22.9\% + 7.9\% + 7.2\%) / 4$)) than on the right (8.4% ($= (7.2\% + 9.5\% + 8.7\% + 8.1\%) / 4$)). Rambo seemed to be robust for all scenarios as MRs showed small values of r_{FDE} ($< 5\%$), except for the scenario with the location right-300, where it was as high as 39.0%. It was revealed from the new MRs that while the ego car was facing a car moving toward it, each system made different decisions in response to the relative difference in positions of the car. The Rambo model behaved more reasonably by having a more correct steering maneuver to avoid the collision in all scenarios. Different unique patterns were also observed with both the Chauffeur and Autumn models (Fig. 3).

Our new MRs also revealed that all ADS models were sensitive to the changing characteristics of objects involved. Whilst the color of the leading car switched from red to white, or red to blue, the Chauffeur model blundered in behaving correctly (with high r_{FDE} values of 21.6% and 36.0%, respectively, as shown in Fig. 3). In comparison, both the Autumn and Rambo models were robust against the color change of the leading car with an effectiveness value of roughly 5.1% or much smaller. When the leading object was either a bus or a bike instead of a car, the r_{FDE} value for the Chauffeur model was also large (21.6%–34.6%), whereas it was smaller in either the Autumn or Rambo model. The scenario where the Autumn model had a larger number of violations than the Chauffeur model was the one associated with the object changing from a dog to a kangaroo, where its effectiveness was 6.3%. In general, the Rambo model was more robust against the object changes than other two models.

2) *Insights into responses to environmental changes:* Existing MRs were helpful in determining whether the SUT is robust in responding to a certain environmental condition. Our new MRs further explored the robustness against the fine-

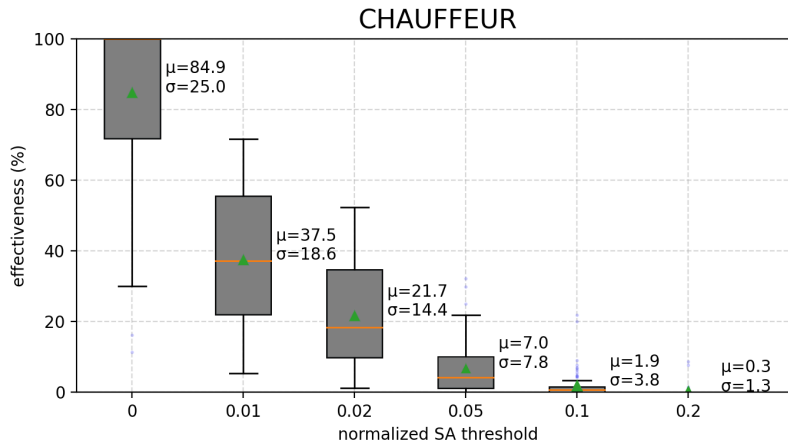


Fig. 1. Effectiveness of all MRs (averaged) in revealing failures in Chauffeur model against different thresholds of undesirable behavior from all MGs. μ and σ are the arithmetic means and standard deviations, respectively.

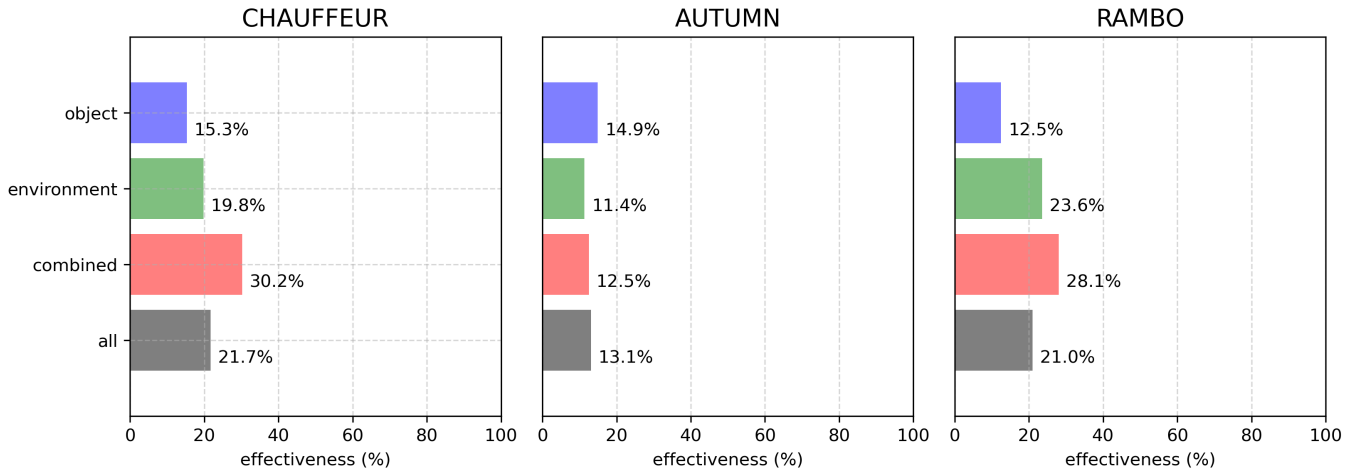


Fig. 2. Failure-detection effectiveness of new MRs in different categories in three models for the same threshold. The blue, green, and red bars denote the performance of MRs for object-based, environment-based, and combined scenarios, respectively; while the grey bar represents the overall performance across all MRs.

gained “changes” in environmental conditions. For instance, existing MRs could guarantee that the SUT works consistently in both sunny day (source test case) and rainy weather (follow-up test case). Our new MRs further assured that the SUT had the robust decisions across different intensities of rains, and gave us insights into the level change in which the system started responding undesirably.

When it came to MRs associated with environmental changes, the responses of three SUTs were different from the above-mentioned object-based MRs. Changing the weather condition from light rain to medium or heavy rain caused the Rambo model to have highly abnormal responses with r_{FDE} as large as of 35.3% or 51.5% (Fig. 3). In comparison, the Autumn model was the best one among three models with the effectiveness as low as 10.3% or 15.1%. In response to gravel roads, the Rambo model was also not very robust, with r_{FDE} ranging from 26.5% to 29.5%. For snowy weather, Rambo

was slightly better (11.5%-16.5%), but still much worse than the Autumn model (1.8%-5.3%).

All three models were more sensitive to the changes of darkness than the brightness conditions according to our new MRs. Note that it is reasonable for images to become darker due to a low-light condition or a degradation of sensor quality over time; whilst the images are brighter due to the influence of a strong light source, making them two separate MRs. For the Chauffeur model, the average failure-detection effectiveness for all brightness-change-related scenarios was $r_{FDE} = 11.5\%$ $(=(2.5\%+6.7\%+13.4\%+23.4\%)/4)$, which was smaller than that of all darkness-change-related scenarios, which was about 13.7% $(=(2.3\%+3.1\%+4.2\%+45.2\%)/4)$. The values for the Autumn model were 7.5% $(=(10.2\%+1.7\%+2.9\%+15.3\%)/4)$ and 19.9% $(=(5.9+9.2+19.3+45.1)/4)$, respectively. The corresponding values for the Rambo models were 4.4% $(=(0.6\%+3.0\%+5.5\%+8.6\%)/4)$ and

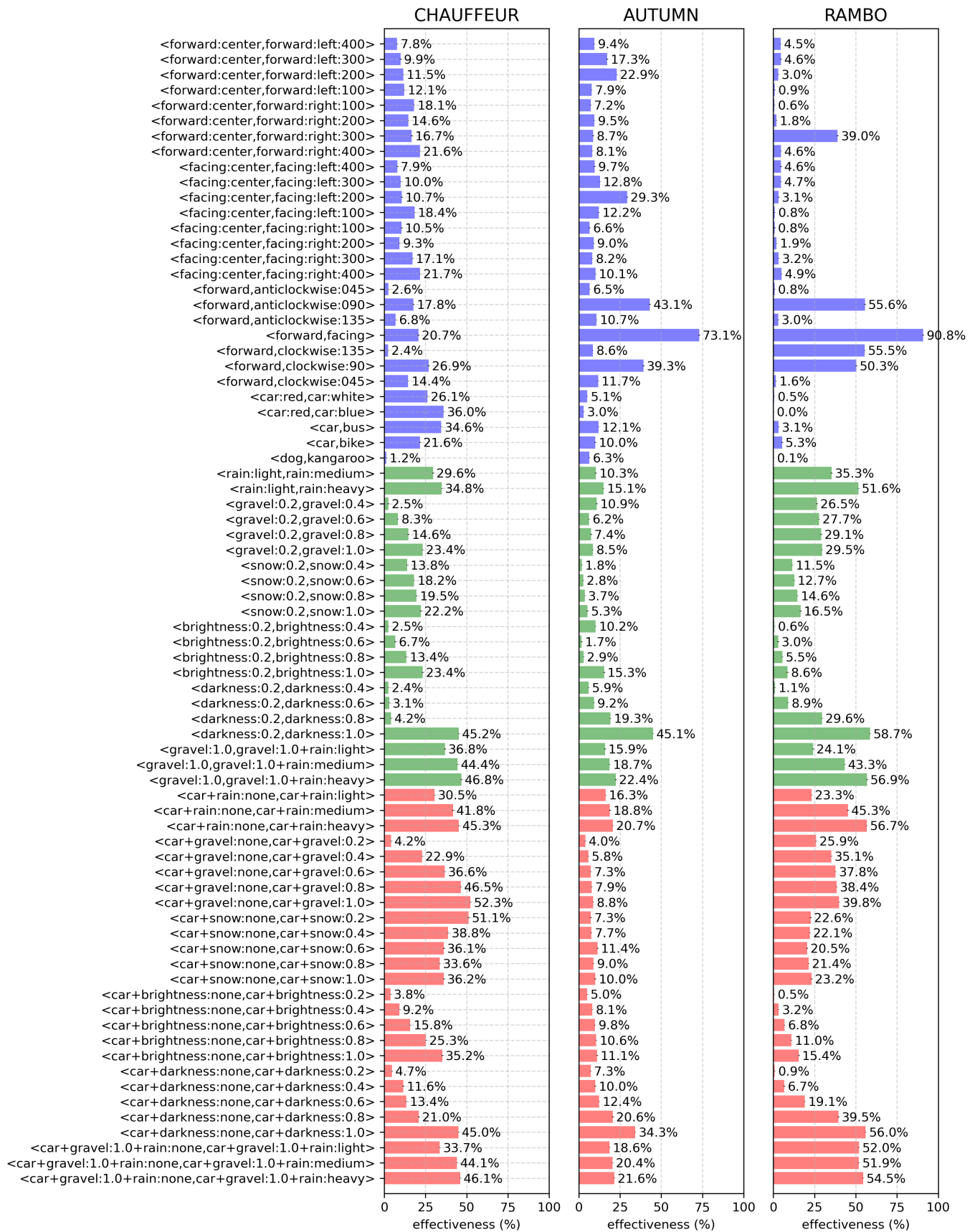


Fig. 3. The average FDE of new MRs for different models for the default threshold ($\theta_n = 0.02$). The blue, green, and red bars represent the average FDE of MRs for object-based, environment-based, and combined scenarios, respectively.

24.6% $(=(1.1\%+8.9\%+29.6\%+58.7\%)/4)$. In a word, the darkness-change-related MRs revealed more failures than the brightness-change-related MRs.

3) *Insights into responses to complex combinations of environmental changes and objects*: The MRs for combined scenarios in general helped detect more issues in the ADS in most cases as compared to the sole environment- or object-related ones. The effectiveness, however, differed significantly among MRs and models. Its average values varied greatly for the Chauffeur model, from 4.2% ($\langle \text{car}, \text{car}+\text{gravel}-0.2 \rangle$) to 52.3% ($\langle \text{car}, \text{car}+\text{gravel}-1.0 \rangle$) (Fig. 3). In comparison, the range of r_{FDE} of new MRs applied to the Autumn model was between 4.0% ($\langle \text{car}, \text{car}+\text{gravel}-0.2 \rangle$) and 34.3% ($\langle \text{car}, \text{car}+\text{darkness}-1.0 \rangle$). For the Rambo model, its lowest r_{FDE} value was 0.5% ($\langle \text{car}, \text{car}+\text{brightness}-0.2 \rangle$) and its highest one was 56.7% ($\langle \text{car}, \text{car}+\text{rain}-\text{heavy} \rangle$). In summary, the MRs for combined scenarios are more effective in failure detection than those for the pure environment- or object-related scenarios.

An interesting finding on the common behavior of these models revealed by our new MRs was that they aligned well with human intuition in most scenarios (except the snow conditions). For example, the heavier the rain was, the less desirable was its behavior. For the Chauffeur model, the new MRs on the changes in scenarios involved a leading car from a normal (sunny) weather to light, medium, and heavy rain have the failure-detection effectiveness of 30.5%, 41.8%, and 34.3%, respectively (Fig. 3). Similar trends were observed in other MRs involving the combined factors related to the change of intensities of the brightness, the darkness, as well as the combined rain and gravel conditions. Such tendencies were also spotted in the Autumn and Rambo models with our new MRs. It would be interesting and worthwhile to investigate how to improve the robustness of these models under various scenarios.

IV. CONCLUDING REMARKS

In this study, we identified completely new sets of MRs and systematically adopted them in a diverse sequential way to test perception systems in autonomous driving by enhancing the SMART framework [6]. Systematic applications of these MRs enabled the detection of more failures and provided deeper insights into the behavior of three selected deep learning systems. Specifically, we showed that:

- Our MRs were highly effective in detecting new failures in different SUTs in all categories of scenarios. For the default threshold, their r_{FDE} values are in the range of 13.1% to 21.7%. It is reasonable to observe that our MRs had different values of FDE for different SUTs and different scenarios.
- The new methodology allowed us to gain deep insights into how the SUT responded to the changes under diverse scenarios in the object-related, the environment-based, and the combined categories for autonomous driving.

Overall, our findings demonstrate that while testing AVs remains constantly of great importance, there are more MRs to be identified. These MRs, if developed systematically, do not only enable us to detect failures with the SUT, but also provide us with much deeper insights.

REFERENCES

- [1] Y. Tian, K. Pei, S. Jana, and B. Ray, "DeepTest: Automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the 40th International Conference on Software Engineering*, ser. ICSE '18. New York, NY, USA: ACM, 2018, pp. 303–314.
- [2] Y. Deng, X. Zheng, T. Zhang, H. Liu, G. Lou, M. Kim, and T. Y. Chen, "A declarative metamorphic testing framework for autonomous driving," *IEEE Transactions on Software Engineering*, 2022.
- [3] P. Kaur, S. Taghavi, Z. Tian, and W. Shi, "A survey on simulators for testing self-driving cars," *arXiv:Robotics*, 2021.
- [4] N. Rajabli, F. Flammini, R. Nardone, and V. Vittorini, "Software verification and validation of safe autonomous cars: A systematic literature review," *IEEE Access*, vol. 9, pp. 4797–4819, 2021.
- [5] Z. Q. Zhou and L. Sun, "Metamorphic testing of driverless cars," *Communications of ACM*, vol. 62, no. 3, pp. 61–67, Feb. 2019.
- [6] Q.-H. Luu, H. Liu, T. Y. Chen, and H. L. Vu, "A sequential metamorphic testing framework for understanding autonomous vehicle's decisions," *IEEE Transactions on Intelligent Vehicles*, pp. 1–13, 2024.
- [7] S. Segura, G. Fraser, A. Sanchez, and A. Ruiz-Cortes, "A survey on metamorphic testing," *IEEE Transactions on Software Engineering*, vol. 42, no. 9, pp. 805–824, Sept 2016.
- [8] T. Y. Chen, F. C. Kuo, H. Liu, P. Poon, D. Towey, T. H. Tse, and Z. Q. Zhou, "Metamorphic testing: A review of challenges and opportunities," *ACM Computing Surveys*, vol. 51, no. 1, 2018.
- [9] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid, "DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems," in *2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2018, pp. 132–142.
- [10] Y. Zhang, D. Towey, M. Pike, J. Cheng Han, Z. Quan Zhou, C. Yin, Q. Wang, and C. Xie, "Scenario-driven metamorphic testing for autonomous driving simulators," *Software Testing, Verification and Reliability*, vol. 34, no. 7, p. e1892, 2024, e1892 stvr.1892. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/stvr.1892>
- [11] Z. Yang, S. Huang, X. Wang, T. Bai, and Y. Wang, "MT-Nod: Metamorphic testing for detecting non-optimal decisions of autonomous driving systems in interactive scenarios," *Information and Software Technology*, vol. 180, p. 107659, 2025.

- [12] Q.-H. Luu, H. Liu, and T. Y. Chen, "Can ChatGPT advance software testing intelligence? An experience report on metamorphic testing," 2023. [Online]. Available: <https://arxiv.org/abs/2310.19204>
- [13] Y. Zhang, T. Y. Chen, M. Pike, D. Towey, Z. Ying, and Z. Q. Zhou, "Enhancing autonomous driving simulations: A hybrid metamorphic testing framework with metamorphic relations generated by GPT," *Information and Software Technology*, vol. 187, p. 107828, 2025.
- [14] Z. Q. Zhou, L. Sun, T. Y. Chen, and D. Towey, "Metamorphic relations for enhancing system understanding and use," *IEEE Transactions on Software Engineering*, vol. 46, no. 10, pp. 1120–1154, 2020.