Collaborative Multi-Agent Testing for Emergent Failure Discovery in Autonomous Driving Systems

Ruizhen Gu
Queen's University Belfast
Belfast, UK
r.gu@qub.ac.uk

Konstantinos Koufos
Queen's University Belfast
Belfast, UK
k.koufos@qub.ac.uk

Donghwan Shin
The University of Sheffield
Sheffield, UK
d.shin@sheffield.ac.uk

Vahid Garousi
Queen's University Belfast
Belfast, UK
Azerbaijan Technical University
Azerbaijan
v.garousi@qub.ac.uk

Mehrdad Dianati
Queen's University Belfast
Belfast, UK
m.dianati@qub.ac.uk

***Abstract*—Autonomous Driving Systems (ADS) can fail because of faults within individual modules as well as from interactions across perception, planning, and control. Yet existing ADS testing research often treats key testing functions, such as perturbation generation, behavioural assessment, and test case selection and exploration, as loosely coupled steps rather than coordinated roles for discovering such failures. We present CREAD, a collaborative multi-agent testing framework for testing ADS that organises perturbation generation, behavioural validation, and search coordination through a shared blackboard and an orchestrator. In the current work-in-progress instantiation, the framework focuses on perception-oriented perturbation generation, while remaining extensible to other ADS modules, including planning and control. It currently comprises a Perception Fuzzer Agent, a Metamorphic Validator Agent, and an Orchestrator Agent. Respectively, they generate perturbations, assess behavioural consistency across related scenario pairs, and coordinate further exploration. Experiments in `HighwayEnv` simulator show that the collaborative configuration improves failure discovery in the highway environment and remains competitive in the roundabout setting. Across the two environments, it yields about 2.1x as many failures per 100 scenarios as the single-agent baseline on average, while gains over a non-collaborative two-agent baseline vary across environments. These results suggest that collaborative multi-agent testing is a promising research direction for emergent ADS behaviour discovery.**

***Index Terms*—autonomous driving systems, software testing, multi-agent systems, large language models.**

I. Introduction

Autonomous Driving Systems (ADS) are safety-critical, open-world systems whose failures can arise both from faults within individual modules and from interactions among perception, planning, and control [1, 2]. Given this complexity, verification and validation for ADS increasingly relies on scenario-based testing to build safety-relevant evidence [3]. In this paradigm, functional scenarios derived from the operational design domain (ODD) are progressively refined into concrete scenarios, which can then be executed as test cases in simulation or other test environments [4]. Within such a pipeline, effective testing requires not only meaningful perturbation generation, but also reliable behavioural assessment and efficient exploration of the scenario space. Yet recent testing approaches based on feedback-guided fuzzing, adaptive search, and LLM-assisted scenario synthesis still struggle to consistently uncover rare but high-consequence failures, especially those arising in the long tail of safety-critical driving scenarios [5–7]. While some failures, such as collisions, are easy to recognise, more subtle behavioural degradations are harder to assess consistently, especially when they emerge from interactions across multiple modules [8].

This work is motivated by two limitations of current ADS testing. First, recent approaches of scenario generation, such as LLM-guided synthesis, improve realism and efficiency, but still struggle to steer exploration towards diverse and safety-relevant scenario families without repeatedly generating similar test cases [1, 6, 7]. Second, current ADS testing pipelines often treat scenario generation and behavioural validation as separate or only loosely coupled stages, limiting feedback between the creation of candidate scenarios and the assessment of their safety relevance [9]. We address these limitations by treating ADS testing as a collaborative process rather than as a single generation-and-evaluation loop.

Drawing on agent-based software testing [10], we frame scenario-based safety-focused testing for ADS as a collaborative testing problem and introduce CREAD (**C**ollabo**R**ative for **E**mergent **A**DS behaviour **D**iscovery). Unlike prior scenario-generation frameworks, including multi-agent and LLM-assisted approaches, which mainly emphasise realistic and efficient construction of challenging scenarios [7, 11], CREAD focuses on coordinating perturbation generation, validation, and exploration to improve failure yield. In the present instantiation, this process is realised through three coordinated roles: (1) Perception Fuzzing Agent for generating perception-oriented perturbations and scenario variants; (2) Metamorphic Validation Agent for comparing baseline

and perturbed executions to detect safety-relevant behavioural inconsistencies; and (3) Orchestrator Agent that prioritises promising scenario families for further testing through a decentralised blackboard architecture. These roles interact through a decentralised blackboard architecture, forming a closed-loop testing process in which agents continually inform one another.

A key advantage of this collaborative design is that it allows complementary testing functions to inform one another during search rather than operating as isolated steps. For example, the Perception Fuzzer may generate a mild-glare motorway-merge scenario, and the Metamorphic Validator may then detect delayed braking or reduced time-to-collision relative to the baseline. This feedback is written to the blackboard and used by the Orchestrator to prioritise similar or nearby scenarios in subsequent explorations.

In our evaluation using `HighwayEnv`, in the Highway environment, the collaborative configuration increases the number of collision-causing scenarios from 14 to 52 per 100 tested scenarios relative to a non-collaborative multi-agent baseline, while matching failure-type diversity and increasing the perception fault rate. In the Roundabout environment, it remains competitive, yielding 52 failures per 100 scenarios compared with 58 for the non-collaborative baseline, while slightly improving failure diversity and perception fault rate. Relative to a single-agent baseline, the gains in failure discovery are also substantial, particularly in Highway.

In this work, perception fuzzing serves as a proof-of-concept instantiation rather than a restriction of the framework. The architecture is designed to support other testing roles, including agents targeting planning and control. The current implementation is evaluated in a controlled simulation, with comprehensive evaluation left to future work. The implementation is available at: https://github.com/ruizhengu/CREAD

This work makes two contributions:

- It introduces CREAD as a collaborative multi-agent testing paradigm for ADS, coordinating distinct testing roles through a shared blackboard and an adaptive orchestrator rather than relying on a single end-to-end search engine.
- It provides empirical evidence that the collaborative testing improves failure discovery over simpler baselines. In our study, the collaborative configuration increases failure discovery from 12 to 52 failures per 100 scenarios in the highway and remains competitive in the roundabout setting, outperforming the single-agent baseline while also revealing environment-dependent tradeoffs against a non-collaborative multi-agent baseline.

II. Background and Related Work

A. ADS Fuzzing

Coverage-oriented and adversarial fuzzing methods have significantly advanced ADS testing by automating the discovery of safety-critical scenarios in large input spaces. AV-Fuzzer introduced search-based violation discovery in simulated driving and showed that guided mutation can effectively expose safety violations [5]. DriveFuzz further improved this line of work by incorporating driving-quality feedback to guide fuzzing towards vulnerable trajectories more efficiently [6]. More recent studies, including LLM-guided scenario generation, highlight the potential of language-driven synthesis for constructing diverse and realistic edge cases [7].

However, these approaches mainly strengthen scenario generation itself and are typically organised as a single generation-and-evaluation loop. This leaves limited support for coordinating multiple testing roles or reusing intermediate evidence across strategies. Our work addresses this gap by treating fuzzing as one component within a collaborative testing architecture for failure discovery.

B. Metamorphic Testing

The oracle problem remains a major bottleneck in ADS validation, especially when complete ground-truth labels are unavailable [12]. Metamorphic testing addresses this by specifying expected consistency relations between transformed but semantically equivalent inputs. DeepRoad is an early representative example that uses environmental transformations to expose inconsistent driving behaviour [13]. More recent work extends this direction through declarative metamorphic frameworks for autonomous driving [14].

These methods are effective for identifying behaviour-level inconsistencies, but they are often used as standalone validation mechanisms. As a result, metamorphic signals usually play a limited role in guiding subsequent test generation. Our work addresses this gap by treating metamorphic validation as an interacting testing role within the overall discovery loop.

C. Multi-Agent ADS Testing

Collaborative and multi-agent approaches have recently gained traction in ADS testing, including multi-agent reinforcement-learning methods for generating adversarial traffic interactions [11]. These approaches can improve the realism and efficiency of critical scenario generation. More broadly, prior surveys argue that ADS assurance requires combining complementary testing paradigms rather than relying on isolated techniques [1, 15].

Existing multi-agent approaches, however, still focus mainly on generation. They provide limited support for collaboration among complementary testing roles such as perturbation generation, behavioural validation, and adaptive coordination. Our work addresses this gap through a blackboard-based architecture in which specialised agents exchange partial evidence and an orchestrator reallocates effort during failure discovery.

III. Approach

This section presents CREAD, a collaborative multi-agent framework for ADS testing. The framework organises scenario perturbation, behavioural validation, and adaptive coordination as interacting testing roles in a closed loop. Figure 1 shows the overall architecture and the interactions among the Orchestrator, the shared blackboard, and the testing agents.

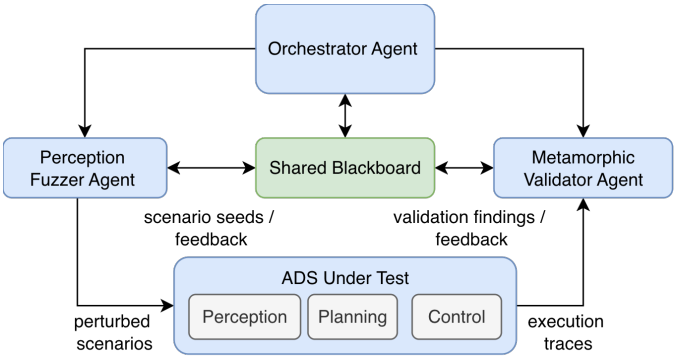


Fig. 1: Overview of CREAD.

At each iteration, the Orchestrator selects or prioritises a baseline scenario and allocates testing effort. The Perception Fuzzer Agent proposes targeted perturbations intended to induce safety-relevant deviations. The simulator executes the baseline and perturbed scenarios, and the Metamorphic Validator Agent compares the resulting behaviours to determine whether the perturbation causes disproportionate behavioural changes. The resulting evidence is written to the blackboard and reused by the Orchestrator to guide subsequent testing.

A. Blackboard Collaboration Mechanism

The blackboard is the framework's shared coordination mechanism. It stores baseline scenarios, perturbed scenarios, execution traces, validation outcomes, and summary coverage information, allowing agents to exchange intermediate evidence asynchronously without direct point-to-point coupling. This shared-state design keeps the agents loosely coupled and makes the framework extensible with additional testing roles.

B. Orchestrator Agent

The Orchestrator Agent coordinates global testing priorities through a coarse QD-inspired archive over scenario and outcome descriptions. Rather than generating perturbations or performing validation itself, it selects which baseline scenarios to revisit or mutate based on heuristic scores derived from prior execution feedback. In the current implementation, these scores combine coarse indicators such as failure yield, severity, novelty, and diversity. Guided by these signals, the orchestrator prioritises seeds from underexplored but promising archive regions and allocates further testing effort to scenario families that appear likely to expose additional failures.

C. Perception Fuzzer Agent

The Perception Fuzzer Agent generates structured test scenarios intended to stress the ADS under test. In the current implementation, it starts from a seeded scenario template. It applies LLM-guided refinement to produce a scenario specification that includes the traffic configuration, the ego vehicle's initial state, adversarial vehicles, and a perception configuration. The perception configuration includes presets such as fog, rain, night, or dusk, which are then mapped to the perturbation model used at execution time.

Each generated scenario is evaluated in two passes. The framework first executes the scenario as a baseline run without perception perturbation, and then executes the same scenario again with the perturbation specified by its perception configuration. This design enables comparisons of behavioural differences under matched scenario conditions, rather than across independently generated scenes.

The agent records each generated scenario, its perturbation context, and the resulting outcomes, shared through the blackboard to support metamorphic validation and later scenario selection by the orchestrator. In this way, the agent functions as one testing role within a collaborative loop, rather than as a standalone scenario generator.

D. Metamorphic Validator Agent

The Metamorphic Validator Agent provides behavioural assessments for the framework. It compares baseline and perturbed executions to determine whether a scenario variation leads to disproportionate changes in safety-relevant behaviour. In the current implementation, the comparison is based on execution-level summaries such as collision, near-miss occurrence, and perception-related faults, where the latter denotes safety-relevant discrepancies between the simulator ground truth and the perturbed perceived state induced by the Fuzzer. A violation is reported when the perturbed run exceeds rule-based thresholds relative to baseline, for example, through increased collision rate, run under the same scenario structure. The resulting signal is therefore an indicator of behavioural inconsistency rather than a complete formal oracle.

Based on this framework, we study two research questions in the evaluation:

- **RQ1:** Does agent collaboration improve failure discovery compared to non-collaborative configurations with the same number of agents?
- **RQ2:** Does a non-collaborative multi-agent configuration improve failure discovery over a single-agent baseline?

IV. Evaluation

This section evaluates whether the proposed collaborative testing architecture improves failure discovery in ADS testing. Our evaluation is designed to address the research questions proposed in Section III. To answer these questions, we conduct an ablation study under controlled simulation conditions.

A. Evaluation Methodology

We conduct an ablation study to isolate the contribution of collaborative and multi-agent architecture. Three configurations are evaluated: (1) **Collaborative (Full System):** Both testing agents (Perception Fuzzer and Metamorphic Validator) with blackboard and orchestration; (2) **Non-collaborative:** Both testing agents without blackboard or orchestration, executed in sequential order; (3) **Single-Agent (Baseline):** Perception Fuzzer agent only, without collaboration infrastructure.

Table I summarises these configurations. The comparison between Collaborative and Non-collaborative isolates the effect of inter-agent collaboration, while the comparison be-

Configuration	Agents	Blackboard	Orchestrator
Collaborative (Full System)	F + M	✓	✓
Non-collaborative	F + M	X	X
Single-Agent (Baseline)	F	X	X

TABLE I: Study configurations (F = Perception Fuzzer Agent, M = Metamorphic Validator Agent). Both Collaborative and Non-collaborative configurations employ the same agents, differing in whether collaboration infrastructure is enabled.

tween Non-collaborative and Single-Agent isolates the contribution of adding multiple testing agents without collaboration.

Each configuration is executed for 50 iterations in each environment. The study is intended to evaluate the testing architecture as a failure-discovery mechanism rather than to benchmark driving quality or controller performance.

B. Experimental Setup

1) Simulator and ADS Under Test: `HighwayEnv` [16] is used as the controlled ablation platform. It provides several compact driving scenarios, such as *Highway*, *Roundabout*, *Merge*, and *Intersection*. We select *Highway* and *Roundabout* because they offer two complementary interaction profiles: a structured straight-road environment and a denser roundabout with richer vehicle interactions.

We choose `HighwayEnv` because it supports efficient repeated execution and direct access to traffic state under reproducible conditions, making it well-suited for comparing testing architectures in a controlled setting. Surrounding traffic follows the simulator's built-in microscopic behaviour models, including IDM-style (intelligent driver model) dynamics [17].

To isolate the effect of the testing architecture, we keep the ADS under test simple and fixed across all ablations. The ego vehicle is controlled by a deterministic heuristic policy supported by `HighwayEnv`[1], selecting lane-change and speed-control actions at each simulation step. For evaluation, each generated scenario is executed twice, once without perturbation and once with the derived perturbation, with each rollout capped at 50 simulation steps. The ego policy should therefore be viewed as a rule-based behaviour-selection baseline rather than a learned controller [18, 19].

2) Perception Perturbation Model: `HighwayEnv` does not provide an explicit perception stack. To study perception-related failures, we model perception degradation as structured corruption of the ego vehicle's perceived state derived from simulator ground truth, including both the ego vehicle's own perceived state and safety-critical cues about nearby traffic. The perturbation module injects safety-relevant distortions such as lane-offset and speed-estimation noise, and position error. These perturbations are then consumed by the ego policy and by the fault-analysis metrics, enabling us to study how perception errors propagate into planning and control behaviour under controlled conditions.

This setup should be interpreted as a state-estimation-level proxy for perception degradation rather than a photorealistic sensor simulation. Its purpose is to support architecture-level ablation of collaborative testing behaviour, not to replace high-fidelity validation in a simulator such as CARLA [20].

3) AI Models: In the current evaluation, the Perception Fuzzer Agent is the only component that uses an LLM. It uses z.ai's `GLM-5-Turbo`[2] through API calls to refine seeded scenarios into targeted perception-oriented test cases. Although the broader framework can accommodate LLM-based reasoning in other agents, those capabilities are not part of the reported ablation setup.

C. Evaluation Metrics

As the goal of the framework is failure discovery rather than driving performance optimisation, our primary metric is *failures per 100 scenarios*, which measures how often a testing configuration exposes failure-causing scenarios. To better characterise the discovered failures, we also report *unique failure types*, which count how many distinct predefined failure categories are triggered during execution. In the current study, these categories are collision, near-miss, lane departure, speed violation, and perception fault (discussed in Section III-D). We further report *perception fault rate*, which reflects sensitivity to simulated perception degradation. Finally, we include *collision rate* and *near-miss rate* as supporting indicators of the severity profile of the discovered scenarios.

D. RQ1: Impact of Inter-Agent Collaboration

Table II summarises the final LLM-enabled ablation results for Highway and Roundabout over 50 iterations per configuration. To assess the effect of inter-agent collaboration, we compare Collaborative and Non-collaborative multi-agent settings, which differ in whether they use blackboard-based coordination and orchestration.

The results show that the effect of collaboration is positive, but it depends on the environment. In Highway, the collaborative configuration substantially improves failure discovery, increasing failures from 14 to 52 per 100 scenarios. It also matches the non-collaborative baseline on unique failure types (3.9 vs. 3.9). In Roundabout, the collaborative configuration does not improve failure discovery, yielding 52 failures per 100 scenarios compared with 58 for the non-collaborative baseline. However, it slightly improves failure diversity (3.8 vs. 3.7).

Overall, these results suggest that blackboard-based coordination can strengthen failure discovery in some environments, but its benefit is not consistent across all settings.

Answer to RQ1: Agent collaboration improves failure discovery in Highway, but not in Roundabout. In Highway, it increases failures from 14 to 52 per 100 scenarios while maintaining the same failure diversity and raising the perception fault rate. In Roundabout, it does not outperform the non-collaborative baseline in failure frequency, although it slightly improves failure diversity and perception fault rate.

[1] https://highway-env.farama.org/dynamics/vehicle/behavior/

[2] https://docs.z.ai/guides/llm/glm-5-turbo

	Highway			Roundabout		
Metric	**Collaborative**	**Non-collaborative**	**Single-Agent**	**Collaborative**	**Non-collaborative**	**Single-Agent**
Failures/100 scenarios	**52.00**	14.00	12.00	52.00	**58.00**	38.00
Unique failure types	**3.9**	**3.9**	3.7	**3.8**	3.7	**3.8**
Collision rate	**12.98%**	1.98%	0.78%	**15.66%**	11.45%	6.78%
Near-miss rate	**44.22%**	28.79%	27.49%	21.16%	**23.92%**	20.03%
Perception fault rate	**20.07%**	14.30%	15.25%	19.08%	15.77%	**19.46%**

TABLE II: LLM-enabled ablation results across the Highway and Roundabout environments (50 iterations each).

E. RQ2: Contribution of Multi-Agent Architecture

We compare the Non-collaborative and Single-Agent configurations to evaluate whether adding multiple testing agents, without collaboration, improves failure discovery over a single-agent baseline.

The results show that the non-collaborative multi-agent configuration improves failure discovery in both environments, although the size of the gain varies. In Highway, failures increase from 12 to 14 per 100 scenarios, and failure diversity also rises slightly from 3.7 to 3.9. However, the perception fault rate decreases slightly from 15.25% to 14.30%. In Roundabout, the improvement in failure discovery is much larger, with failures increasing from 38 to 58 per 100 scenarios. The two configurations remain similar in failure diversity (3.7 vs. 3.8), while the perception fault rate is lower in the non-collaborative multi-agent setting (15.77% vs. 19.46%).

Overall, these results suggest that adding multiple agents can improve failure discovery even without collaboration, especially in the Roundabout environment, although the gains are less consistent across the other metrics.

> **Answer to RQ2:** The non-collaborative multi-agent configuration improves failure discovery over the single-agent baseline in both environments. In Highway, failures increase from 12 to 14 per 100 scenarios, while in Roundabout, they increase more substantially from 38 to 58. However, improvements in failure diversity and perception fault rate are less consistent.

F. Discussion

The evaluation shows that the proposed architecture is promising as a failure-discovery framework, but that its advantages depend on the traffic environment and on the comparison baseline. Inter-agent collaboration shows benefit in Highway, which substantially increases failure yield while maintaining failure diversity and increasing perception fault rate. In Roundabout, collaboration remains competitive, but does not dominate the non-collaborative multi-agent configuration on the primary failure-discovery metric.

A second observation is that multiple testing roles are beneficial even without collaboration. The comparison between Non-collaborative and Single-Agent shows that adding a testing role can improve failure discovery, particularly in Roundabout. This suggests that the architectural decomposition of testing functions already contributes value, while collaboration further shapes how effectively those roles interact.

Overall, the results support collaborative testing as a useful architectural direction for ADS failure discovery, while also indicating that interaction-heavy environments remain more sensitive to coordination strategy and search design.

G. Limitations

Perception Abstraction: `HighwayEnv` does not provide an explicit full sensor simulation, such as a camera or LiDAR. To study perception-related failures, we approximate perception degradation by injecting structured perturbations into the ego vehicle's derived perceived state. This allows controlled analysis of how perception errors propagate into downstream behaviour, but does not reflect a full-stack perception.

Controlled Ablation Platform: `HighwayEnv` is a lightweight traffic simulator and does not provide the realism of higher-fidelity ADS platforms such as CARLA [20]. We use it as a controlled ablation environment to enable fast, repeatable evaluation of the testing architecture, but the external validity of the results remains limited. Extending the framework to higher-fidelity simulators such as CARLA is part of our future work.

Heuristic Ego Policy: The ego vehicle is controlled by a deterministic heuristic policy rather than a production ADS stack or a learned end-to-end controller. We use this simplified policy to isolate the effect of the testing architecture, but the resulting failures should be interpreted as evidence of architectural behaviour under controlled conditions rather than as realistic deployed ADS failures.

ODD-Constrained Perturbations: In the current version of CREAD, perturbations are designed to remain within the operational design domain (ODD) of the underlying driving setting. This allows us to study whether safety-relevant failures can emerge under plausible in-domain variation, rather than from explicitly out-of-domain conditions. As a result, the current evaluation does not yet assess how the framework behaves when scenarios cross ODD boundaries or expose the ADS to clearly unsupported conditions.

V. Conclusion and Future Work

This paper presented CREAD, a collaborative multi-agent testing framework for Autonomous Driving Systems. It organises perturbation generation, behavioural validation, and coordination as interacting testing roles, rather than treating ADS testing as a single generation-and-evaluation loop. Our

evaluation in `HighwayEnv` shows that the benefits of collaboration are positive but environment-dependent. In the Highway environment, the collaborative configuration improves failure discovery over the non-collaborative multi-agent baseline, increasing failures from 14 to 52 per 100 scenarios. In Roundabout, it remains competitive but does not outperform the non-collaborative baseline. We also find that the non-collaborative multi-agent configuration improves failure discovery over the single-agent baseline in both environments. Overall, these results suggest that collaborative testing is a promising direction for ADS failure discovery.

Future work will extend the framework in three directions. First, we will evaluate it in higher-fidelity simulators such as CARLA to assess its effectiveness under more realistic sensing and traffic conditions. Second, we will incorporate more advanced ADS controllers, including learning-based driving agents, to study the framework beyond the current heuristic ego policy. Third, we will expand the evaluation to additional driving scenarios and conduct a more comprehensive empirical study of the framework's strengths and limitations.

Acknowledgement

This work constitutes a part of the HIVEMIND project funded by the European Commission under the Horizon Europe call HORIZON-CL4-2024-DIGITAL-EMERGING-01 under Grant Agreement Number 101189745.

References

[1] G. Lou, Y. Deng, X. Zheng, M. Zhang, and T. Zhang, "Testing of autonomous driving systems: where are we and where should we go?" in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2022. ACM, 2022.

[2] S. D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, D. Rus, and M. H. Ang, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, 2017.

[3] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, "Survey on scenario-based safety assessment of automated vehicles," *IEEE Access*, 2020.

[4] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for development, test and validation of automated vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[5] G. Li, Y. Li, S. Jha, T. Tsai, M. Sullivan, S. K. S. Hari, Z. Kalbarczyk, and R. Iyer, "AV-FUZZER: Finding safety violations in autonomous driving systems," in *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*, 2020.

[6] S. Kim, M. Liu, J. J. Rhee, Y. Jeon, Y. Kwon, and C. H. Kim, "Drivefuzz: Discovering autonomous driving bugs through driving quality-guided fuzzing," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022.

[7] S. Tang, Z. Zhang, J. Zhou, L. Lei, Y. Zhou, and Y. Xue, "Legend: A top-down approach to scenario generation of autonomous driving systems assisted by large language models," in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '24. ACM, 2024.

[8] F. Molina, A. Gorla, and M. d'Amorim, "Test oracle automation in the era of llms," *ACM Transactions on Software Engineering and Methodology*, 2025.

[9] Q. Song, H. Ye, M. Harman, and F. Sarro, "Generative ai for testing of autonomous driving systems: A survey," *ACM Trans. Softw. Eng. Methodol.*, 2026, just Accepted.

[10] J. Wang, Y. Huang, C. Chen, Z. Liu, S. Wang, and Q. Wang, "Software testing with large language models: Survey, landscape, and vision," *IEEE Transactions on Software Engineering*, 2024.

[11] T. Wang, S. Tian, G. A. Minas, and C. Bian, "Amacollision: An advanced framework for testing autonomous vehicles based on adversarial multi-agent," *Journal of Systems and Software*, 2025.

[12] G. Jahangirova, A. Stocco, and P. Tonella, "Quality metrics and oracles for autonomous vehicles testing," in *2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST)*, 2021.

[13] 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 *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, 2018.

[14] 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*, 2023.

[15] W. Ding, C. Xu, M. Arief, H. Lin, B. Li, and D. Zhao, "A survey on safety-critical driving scenario generation—a methodological perspective," *Trans. Intell. Transport. Sys.*, 2023.

[16] E. Leurent, "An environment for autonomous driving decision-making," https://github.com/eleurent/highway-env, 2018.

[17] A. Kesting, M. Treiber, and D. Helbing, "Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2010.

[18] A. Kesting, "Mobil : General lane-changing model for car-following models," 2007. [Online]. Available: https://api.semanticscholar.org/CorpusID:19712367

[19] O. Sharma, N. Sahoo, and N. Puhan, "Recent advances in motion and behavior planning techniques for software architecture of autonomous vehicles: A state-of-the-art survey," *Engineering Applications of Artificial Intelligence*, 2021.

[20] 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*, 2017.