Reinforcement Learning for Falsification of Dynamic Driving Scenarios

Extended Abstract

Oliver Chang
University of California, Santa Cruz
Santa Cruz, CA, United States
elochang@ucsc.edu

Kay Vargas
University of California, Santa Cruz
Santa Cruz, CA, United States
kvarga12@ucsc.edu

Leilani Gilpin
University of California, Santa Cruz
Santa Cruz, CA, United States
lgilpin@ucsc.edu

Daniel J. Fremont
University of California, Santa Cruz
Santa Cruz, CA, United States
dfremont@ucsc.edu

ABSTRACT

Falsification has been widely used to find failure cases for cyber-physical systems (CPS). In the domain of autonomous driving, falsification has recently been applied to find adversarial driving maneuvers which cause other vehicles to crash. In this work, we propose a reinforcement learning (RL)-based falsification framework that can discover complex adversarial maneuvers in diverse driving scenarios. We compare our approach to existing falsification methods, both in terms of their efficiency at finding counter-examples as well as the diversity and quality of their counter-examples. Our results suggest that RL-based falsification can be an effective tool for testing and validating autonomous vehicle systems.

KEYWORDS

Reinforcement Learning; Falsification; Simulation

ACM Reference Format:
Oliver Chang, Kay Vargas, Leilani Gilpin, and Daniel J. Fremont. 2026. Reinforcement Learning for Falsification of Dynamic Driving Scenarios: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/PFBJ1092

1 INTRODUCTION

Autonomous driving has shown significant progress in recent years. While autonomous driving CPS work under ideal conditions, real-world scenarios often introduce disturbances that can affect system performance [5]. Ensuring the safety and reliability of these systems is crucial, as failures can have catastrophic consequences. Sampling-based methods, such as cross-entropy (CE) [2] and Multi-armed bandit (MAB) [14], have been widely used for falsification. However, sampling-based search techniques are limited by their ability to explore the input space effectively, especially in high-dimensional spaces, which could result in convergence to local minima in black-box optimization.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/PFBJ1092

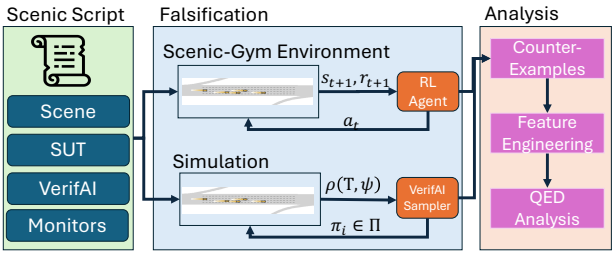

Figure 1: Overview of our falsification framework.

Our work addresses these limitations by introducing an RL-based falsification framework that can generate diverse and meaningful counter-examples across a variety of driving scenarios, including those involving lateral dynamics. Prior work on falsification to find adversarial maneuvers focused on single-lane highway scenarios, only considering longitudinal dynamics to stress test ACC [5, 9, 13]. Moreover, we train a soft actor-critic (SAC) agent to learn a continuous control policy, whereas prior work has used discrete action spaces or low-fidelity observation spaces [6, 9, 13].

We demonstrate our approach by stress-testing adaptive cruise control (ACC) equipped with lane-changing (MOBIL) [10]. We compare our RL-based approach against sampling-based methods (CE) using quality, efficiency, and diversity (QED) metrics to evaluate the discovered counter-examples. Our results suggest that RL-based falsification can be an effective tool for testing and validating AV systems. In summary, our contributions are as follows:

- We introduce a RL-based falsification framework that leverages Scenic [3] to create diverse scenarios.
- We optimize ScenicGym [12] to increase the throughput of samples, which allows us to find more failure points in AV systems in customized driving scenarios.
- We systematically analyze our derived counter-examples by using quality, efficiency, and diversity metrics, showing that SAC discovers more realistic counter-examples than CE.

2 METHODOLOGY

Metric Temporal Logic (MTL) is a logic that extends Linear Temporal Logic and can be used to reason about system properties over

time [7]. We use MTL to formalize the specification of our attacker. The robustness value, ρ ($\rho : X \times MTL \rightarrow \mathbb{R}$) provides a measure for "how close" the attacker is to inducing a system failure. Given $\mathcal{T}$ and an MTL specification (ψ), robustness returns a real value where $\rho(\mathcal{T}, \psi) > 0 \iff \mathcal{T} \models \psi$. Following Hernandez et al. [5], the attacker is successful if it induces an accident among the victim vehicles while remaining safe. More formally, $\bar{\psi} = \varrho_{\text{adv}} \wedge \varrho_{\text{safe}}$, where $\varrho_{\text{adv}} = F \bigvee_{i=1}^{N-1} \neg\varphi_i$ and $\varrho_{\text{safe}} = G\varphi_0$ [5]. ϱ_{adv} specifies that at some point, a victim vehicle's safety specification (φ_i) must be violated. φ_i is the atomic proposition defined as ($d_{i,i+1} > d_{\text{safe}}$), where d_{safe} is the safety distance between any two vehicles. ϱ_{safe} means the adversary's specification φ_0 must always hold true.

We model our RL agent as the adversarial vehicle in the scenario. The RL agent's goal is to learn a policy, $\pi_\theta(a_t|s_t)$, that generates throttle and steer commands that violate ψ. The state space, $\mathcal{S}$, is a stacked vector of size 346, containing LiDAR point-cloud data from multiple LiDAR scanners. The action space, $\mathcal{A}$, consists of continuous control inputs for acceleration and steering commands.

Equation 1 summarizes our reward function in three conditions.

$$r_t = \begin{cases} r_{termination=success} & \text{if } \rho(\mathcal{T}, \bar{\psi}) > 0 \text{ and } t = T \\ r_{termination=fail} & \text{if } \rho(\mathcal{T}, \bar{\psi}) \leq 0 \text{ and } t = T \\ c_1 r_{forward} + c_2 r_{speed} & \text{if t < T} \end{cases} \quad (1)$$

where $r_{forward} = \text{ego}_{x_i} - \text{ego}_{x_{i-1}}$ and $r_{speed} = \frac{\text{ego}_{\text{speed}}}{\text{ego}_{\text{max speed}}}$. We use SAC [4] to find adversarial driving maneuvers. We train a SAC agent on two environments: a single-lane platoon scenario and a multi-lane highway scenario. Both environments place the adversary in front of the victim vehicles. We compare SAC against CE. Figure 1 illustrates our framework to compare these two falsification paradigms. Given a Scenic script containing a scenario definition, map specification, and driving behavior, we use SAC and CE to find counter examples in the system-under-test (SUT).

Our ScenicGym interface follows [1] and [12] with the difference being that we instantiate the simulator once. To show our ScenicGym's increased throughput, we conduct an experiment where we measure the Steps Per Second (SPS) of our ScenicGym interface. On eight parallel sub-environments, we measure the SPS over 100,000 timesteps across three different map sizes consisting of 84, 200, and 323 road units, averaged over four seeded runs. In each map size, our ScenicGym version shows an increase in SPS by 82.8%, 47.8%, and 6.00% for the 84, 200, and 323 road unit maps, respectively.

3 RESULTS

Efficiency: Here we denote the corresponding effectiveness and computational cost associated with each attack. Table 1 contains results for each measure. For efficiency the average number of counter-examples found per attack is displayed along with the average time per simulation (TPS).

Coverage: We quantify coverage using dispersion and discrepancy metrics as in prior work [2, 11]. Each metric is defined in terms of a feature vector; different features are considered for each attack scenario and metric. Features for platoon include: unique crash pairings between vehicles, time until collision, proportion of hard accelerations, and average pairwise distance between all vehicles. Multi-Lane features include: proportion of hard steers/accelerations, unique crash pairings between vehicles, and velocity at impact

Counter-Example Counts		
Platoon	Total	TPS (s)
Cross Entropy	527.75 ± 40.917	3.6
Soft Actor Critic	103.75 ± 49.206	7.3
Multi-Lane	Total	TPS (s)
Cross Entropy	117.75 ± 6.41	2.2
Soft Actor Critic	154.75 ± 7.984	7.2
Coverage Scores by Metric		
Platoon	Discrepancy	Dispersion
Cross Entropy	0.111 ± .001	0.954 ± .003
Soft Actor Critic	0.094 ± .007	0.903 ± .016
Multi-Lane	Discrepancy	Dispersion
Cross Entropy	0.092 ± .023	0.852 ± 0.014
Soft Actor Critic	0.147 ± .01	0.83 ± 0.012
Quality Scores by Attack Type		
Platoon	Risk	KL-Divergence
Cross Entropy	0.174 ± 0.037	1.067 ± 0.767
Soft Actor Critic	0.113 ± 0.036	0.883 ± 0.614
Multi-Lane	Risk	KL-Divergence
Cross Entropy	0.233 ± 0.015	0.087 ± 0.15
Soft Actor Critic	0.402 ± 0.064	0.009 ± 0.171

Table 1: Averaged score for each metric with associated standard deviation over 4 iterations measured over 1000 samples.

Lower scores in both of these categories correspond to better coverage.

Quality: Quality here is measured using the average QD-score as defined in [8] along with the KL-divergence measured from a non-adversarial actor. Table 1 Section 3 shows the computed risk (QD-score) and KL-Divergence averaged over each unique seed. Risk features consist of: the minimum distance between any two victims at the time of collision, the maximal braking at collision time, and the averaged velocity of the two vehicles upon colliding. For KL-Divergence features for platoon include: proportion of hard accelerations and brakes, cumulative time spent outside of a lane (adversary), and the average pairwise robustness between the adversary and the victim cars. For the multi-lane highway scenario the same features are considered, however, the proportion of hard steering action is considered rather than brakes. A higher score and risk and lower score in divergence correspond to a greater overall quality rating.

4 CONCLUSION

We find that SAC is more sample efficient (154 examples) with respect to CE (117 examples) in the multi-lane experiment, but does worse in the platoon scenario. This may be explained, in part, by the scenario's specificity and the template's behavioral bias. However, SAC does perform slightly better in terms of coverage (Discrepancy: 0.094, Dispersion: 0.903), with CE scoring (0.111, 0.954). In total these results show SAC's ability to discover counter-examples in both experiments while forgoing the sampler's template. In sum, our work applies two distinct falsification paradigms, evaluates each approach across multiple dimensions, and opens discussion about the trade-offs between RL and sampling-based falsification.

REFERENCES

[1] Abdus Salam Azad, Edward Kim, Qiancheng Wu, Kimin Lee, Ion Stoica, Pieter Abbeel, Alberto Sangiovanni-Vincentelli, and Sanjit A. Seshia. 2022. Programmatic Modeling and Generation of Real-Time Strategic Soccer Environments for Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 6 (June 2022), 6028–6036. https://doi.org/10.1609/aaai.v36i6.20549 Number: 6.

[2] Anthony Corso, Robert Moss, Mark Koren, Ritchie Lee, and Mykel Kochenderfer. 2021. A Survey of Algorithms for Black-Box Safety Validation of Cyber-Physical Systems. *Journal of Artificial Intelligence Research* 72 (Oct. 2021), 377–428. https://doi.org/10.1613/jair.1.12716

[3] Daniel J. Fremont, Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Alberto L. Sangiovanni-Vincentelli, and Sanjit A. Seshia. 2019. Scenic: a language for scenario specification and scene generation. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*. ACM, Phoenix AZ USA, 63–78. https://doi.org/10.1145/3314221.3314633

[4] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. Pmlr, 1861–1870.

[5] Carlos Hernandez, Diego Ortiz Barbosa, Zengxiang Lei, Luis Burbano, Younghee Park, Satish Ukkusuri, and Alvaro A. Cardenas. 2024. D4: Dynamic Data-Driven Discovery of Adversarial Vehicle Maneuvers. In *International Conference on Dynamic Data Driven Applications Systems*. Springer, 182–190.

[6] Mark Koren, Saud Alsaif, Ritchie Lee, and Mykel J. Kochenderfer. 2018. Adaptive Stress Testing for Autonomous Vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. 1–7. https://doi.org/10.1109/IVS.2018.8500400 ISSN: 1931-0587.

[7] Ron Koymans. 1990. Specifying real-time properties with metric temporal logic. *Real-time systems* 2, 4 (1990), 255–299.

[8] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. 2016. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI* 3 (2016), 40.

[9] Xin Qin, Nikos Arechiga, Jyotirmoy Deshmukh, and Andrew Best. 2023. Robust Testing for Cyber-Physical Systems using Reinforcement Learning. In *Proceedings of the 21st ACM-IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE '23)*. Association for Computing Machinery, New York, NY, USA, 36–46. https://doi.org/10.1145/3610579.3611087

[10] Martin Treiber and Arne Kesting. 2013. *Traffic flow dynamics*. Vol. 1. Springer.

[11] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17 (2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2

[12] Kai Xu. 2025. *ScenicGym: Reinforcement Learning with Data Generation Using Scenic*. Master's thesis. EECS Department, University of California, Berkeley. http://www2.eecs.berkeley.edu/Pubs/TechRpts/2025/EECS-2025-168.html

[13] Yoriyuki Yamagata, Shuang Liu, Takumi Akazaki, Yihai Duan, and Jianye Hao. 2021. Falsification of Cyber-Physical Systems Using Deep Reinforcement Learning. *IEEE Transactions on Software Engineering* 47, 12 (Dec. 2021), 2823–2840. https://doi.org/10.1109/TSE.2020.2969178 Conference Name: IEEE Transactions on Software Engineering.

[14] Zhenya Zhang, Ichiro Hasuo, and Paolo Arcaini. 2019. Multi-armed bandits for boolean connectives in hybrid system falsification. In *International Conference on Computer Aided Verification*. Springer, 401–420.