

USENIX

THE ADVANCED COMPUTING
SYSTEMS ASSOCIATION

WIP: A Benchmark for Security Evaluation of Adversarial Manoeuvre Attacks in Autonomous Driving

Adibeh Emamy, Marcello Maugeri, and PingChen Lin, *CybPass Limited;*
Aryan Pasikhani, *The University of Sheffield*

https://www.usenix.org/conference/vehiclesec26/presentation/emamy

This paper is included in the Proceedings of the
4th USENIX Symposium on Vehicle Security and Privacy.

August 10–11, 2026 • Baltimore, MD, USA

ISBN 978-1-939133-56-4

Open access to the Proceedings of the
4th USENIX Symposium on Vehicle Security and Privacy
is sponsored by USENIX.



WIP: A Benchmark for Security Evaluation of Adversarial Manoeuvre Attacks in Autonomous Driving

Adibeh Emamy
CybPass Limited

Marcello Maugeri
CybPass Limited

PingChen Lin
CybPass Limited

Aryan Pasikhani
The University of Sheffield

Abstract

This paper presents a simulation-only benchmark for security evaluation of autonomous driving agents under adversarial manoeuvre attacks. The benchmark is implemented in *MetaDrive* and supports training attacker policies, evaluating victim policies, and comparing attack outcomes under controlled driving settings. The attacker is black-box with respect to the victim and operates only through physically valid driving actions; sensor spoofing, communication tampering, model-internal access, and code tampering are excluded.

The benchmark records training logs, per-episode outcomes, aggregate safety metrics, trained policies, and evaluation logs for reproducible analysis. Preliminary results show that the benign driving performance of victim policies varies substantially across policy types and driving settings. In addition, highway experiments show that a trained PPO attacker increases the victim mission-failure rate by 10 percentage points. The benchmark provides an open-source platform for training attackers, testing victim policies, and studying adversarial manoeuvre attacks under controlled simulation conditions.

1 Introduction

Physical-world attacks against Autonomous Driving (AD) perception have shown that external perturbations can propagate from sensing errors to unsafe driving outcomes, including attacks on camera-based lane centring [12], multi-sensor fusion [3], and LiDAR-based object detection [2]. Adversarial manoeuvre attacks expose a related security problem: another road user may induce harmful outcomes through physically plausible driving behaviour rather than sensor or perception-channel manipulation [15]. This problem matters because an attacker without privileged software access may still affect the victim through ordinary vehicle control alone.

This paper addresses that gap through a benchmark for security evaluation of robustness to adversarial manoeuvre attacks. The benchmark evaluates two-agent interaction in *MetaDrive*, where victim and attacker vehicles act through shared simulator dynamics. The attacker is black-box, physical-interaction-only, and acts through standard low-level vehicle control. Accordingly, the evaluation measures degradation in victim safety outcomes. The benchmark defines the threat model, simulator interface, controlled factors, outputs, and metrics needed to compare adversarial manoeuvre evaluations across victim policies, driving settings, attacker spawn presets, and attacker designs.

Existing manoeuvre-level attacks in *CARLA* provide the closest attack-oriented reference point [15]. Safety-critical scenario generation in *MetaDrive* provides a related simulator reference point for producing difficult traffic interactions and robustness-oriented training conditions [20]. In parallel, Reinforcement Learning (RL) surveys for AD security emphasise persistent fragmentation in threat modelling, attack/defence categorisation, and reporting methodology [5]. The present benchmark addresses this methodological need through an explicit threat model, a reproducible evaluation protocol, and structured reporting.

The benchmark is motivated by four research questions:

RQ1. To what extent can a physical-interaction-only attacker degrade victim safety outcomes?

RQ2. How does robustness to adversarial manoeuvre attacks vary across DQN, DDQN, PPO, and SAC victims?

RQ3. How does victim robustness to adversarial manoeuvre attacks vary across attacker types, including learned RL and scripted heuristics?

RQ4. To what extent can learned attackers increase attack effectiveness while remaining within realistic behavioural constraints?

As a work in progress, this paper makes the following contributions toward answering these research questions:

- A simulation-only benchmark for security evaluation of robustness to adversarial manoeuvre attacks under a black-box, physical-interaction-only attacker model, with explicit capabilities and constraints.
- A coupled Partially Observable Markov Decision Process (POMDP) formulation for victim and attacker agents over

shared dynamics, with explicit definitions of state, observation, action, reward, and objectives.
- A reproducible benchmark specification and evaluation protocol for training attackers, evaluating victim policies, and recording per-episode outcomes, aggregate safety metrics, trained policies, and evaluation logs.
- Preliminary results showing that benign driving performance varies across victim policies and driving settings, and that a trained PPO attacker increases the victim mission-failure rate by 10 percentage points in the highway setting.

2 Related Work and Positioning

ACERO [15] introduced adversarial driving manoeuvres as a security threat against AD software and evaluated them in *CARLA*. The present benchmark builds on this threat model, but shifts the focus from attack discovery against concrete AD stacks to repeatable evaluation under a constrained black-box attacker model.

Within the same simulator family as the present benchmark, *CAT* [20] generates safety-critical traffic scenarios in *MetaDrive* and uses them to improve policy robustness through adversarial training. *SEAL* [16] and *SAGE* [10] extend this direction by studying generation quality, attacker realism, and controllable trade-offs between adversariality and realism. These works inform the design space, but their primary object remains scenario generation or adversarial training [14] rather than benchmark reporting for a fixed adversarial manoeuvre threat model.

Scenario datasets and generation tools provide reusable inputs for AD simulation testing: *SCTrans* [4] constructs a public scenario dataset with safety-critical traffic interactions, while *ChatScene* [19] generates safety-critical scenarios from structured knowledge and language-guided specifications. *SafeBench* [18] and *MetAdv* [8] move closer to benchmark infrastructure by supporting comparable safety or adversarial testing pipelines. The present benchmark instead fixes the attacker capability model, excluded attack channels, victim outcomes, controlled factors, and reporting outputs for black-box adversarial manoeuvre evaluation.

3 Problem Formulation

The adversarial manoeuvre evaluation problem is formulated as a two-agent interaction between a victim vehicle and an attacker vehicle in the same road scene. A frozen victim policy and an attacker policy act through standard vehicle controls, such as steering, throttle, and braking. At each step, both agents receive observations, select actions, and the road scene evolves according to both actions.

Let $\mathcal{S}$ denote the state space of the road scene, and let $s_t \in \mathcal{S}$ be the state at time t. The victim and attacker receive observations $o_t^v \in O_v$ and $o_t^a \in O_a$, and produce actions $a_t^v \in \mathcal{A}_v$ and $a_t^a \in \mathcal{A}_a$. The victim policy is denoted by $\pi_v(a_t^v \mid o_t^v)$ and is frozen during adversarial evaluation, while the attacker policy is denoted by $\pi_a(a_t^a \mid o_t^a)$.

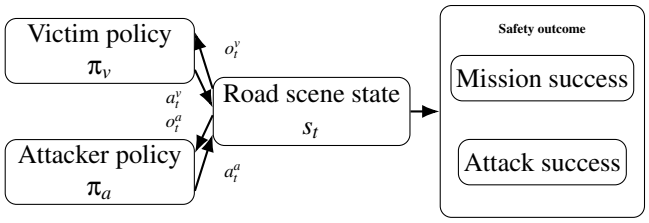


Figure 1: Overview of the attacker-victim interaction.

The shared state-transition kernel is denoted by P, following standard POMDP notation [1, 7]. The two-agent interaction induces transitions of the form

$$P(s_{t+1} \mid s_t, a_t^v, a_t^a). \tag{1}$$

The attacker-victim interaction is coupled because both agents act on the same road scene, i.e., the victim and attacker actions jointly determine the next-state transition.

To formalise partial observability explicitly, the victim and attacker are modelled as coupled Partially Observable Markov Decision Process (POMDP) agents over shared dynamics:

$$\mathcal{M}_v = (\mathcal{S}, \mathcal{A}_v, O_v, P, \Omega_v, r_v, \gamma), \quad \mathcal{M}_a = (\mathcal{S}, \mathcal{A}_a, O_a, P, \Omega_a, r_a, \gamma). \tag{2}$$

Here, Ω_v and Ω_a are observation models, r_v and r_a are the victim and attacker reward functions, and γ is the discount factor.

3.1 Threat Model

The threat model considers one victim vehicle and one attacker vehicle in the same road scene. The single-attacker assumption keeps attribution of safety outcomes explicit.

The attacker is black-box with respect to the victim: it does not access victim model weights, gradients, reward internals, training data, or privileged identifiers. The attacker observes only information available through its own driving observation and acts only through physically valid vehicle controls, such as steering, throttle, and braking. Sensor spoofing, communication tampering, map tampering, model-internal access, and code tampering are excluded. The attacker is constrained by road topology and vehicle dynamics.

3.2 Attacker Objective

The formulation separates the victim mission from the attacker objective. Victim mission success corresponds to reaching the destination without going off-road or colliding. At the same time, for a fixed victim policy, the attacker objective is to maximise the probability that the victim goes off-road or

collides without direct attacker contact:

$$\max_{\pi_a} \Pr_{\pi_v,\pi_a,P}(S=1). \tag{3}$$

Notably, victim collisions with the attacker are excluded from attack success because they involve direct attacker contact rather than indirect victim degradation. Let C_t denote a victim collision that does not involve direct attacker-victim contact. The per-step violation indicator used for attack success is therefore

$$F_t = C_t \vee \mathbb{1}[\text{victim off-road}]. \tag{4}$$

At episode level, the attacker success indicator is defined as

$$S = \mathbb{1}[\exists t : F_t = 1]. \tag{5}$$

Over N episodes, the attack success rate (ASR) is defined as

$$\text{ASR} = \frac{1}{N}\sum_{i=1}^{N} S_i. \tag{6}$$

Figure 1 illustrates this formulation.

TTC is not itself a mission violation, but a constant-velocity estimate of how soon the victim and attacker would collide if their current relative motion continued. It is used as a reward signal during attacker training and as an evaluation signal when reporting how close an interaction came to collision. Let p_t^v and p_t^a denote victim and attacker positions, and let v_t^v and v_t^a denote their velocity vectors. Define the relative position and velocity as $\Delta p_t = p_t^v - p_t^a$ and $\Delta v_t = v_t^v - v_t^a$. For non-zero separation, let c_t denote the projected closing rate, obtained by projecting the relative velocity onto the line joining the victim and attacker:

$$c_t = -\frac{\Delta v_t \cdot \Delta p_t}{\|\Delta p_t\| + \varepsilon}. \tag{7}$$

If c_t is non-positive, the projected distance is not decreasing and no finite collision time is predicted, so TTC is set to ∞. Otherwise,

$$\text{TTC}_t = \frac{\|\Delta p_t\|}{c_t}. \tag{8}$$

The attacker reward used for training follows this objective with terminal victim outcomes, TTC, victim reactions, and attacker failure penalties, as detailed in Section 4.

4 Benchmark Design

This section specifies how the benchmark instantiates the problem formulation as a reproducible experimental instrument. Building on *MetaDrive* enables procedural road generation, configurable traffic density, repeatable scenario seeds, multi-agent vehicle interaction, and direct access to simulator event flags such as arrival, off-road, and collision events. In addition, the benchmark inherits the scope of the selected *MetaDrive* environments, vehicle dynamics, observation interfaces, and event definitions. Figure 2 provides an overview of the benchmark design.

Scenario	Environment class	Road geometry characteristics
Highway	`MultiAgentMetaDrive`	Straight multi-lane segment with low interaction complexity.
Intersection	`MultiAgentIntersectionEnv`	Four-way conflict zone with crossing flows.
Roundabout	`MultiAgentRoundaboutEnv`	Curved merge/yield dynamics with multi-entry interception opportunities.
Bottleneck	`MultiAgentBottleneckEnv`	Lane-drop forced merge and congestion coupling.

Table 1: Scenario set and road geometry characteristics.

Scenarios and interfaces The benchmark is implemented in *MetaDrive* and supports four road geometries: highway, intersection, roundabout, and bottleneck. Table 1 summarises the scenario, the environment class, and the road geometry characteristics.

Victim policies receive the observation exposed by the selected *MetaDrive* multi-agent environment class. In the highway setting, this observation is provided by `MultiAgentMetaDrive`. It contains ego-state features describing the controlled vehicle and LIDAR features describing nearby road context and obstacles. The attacker receives the same raw *MetaDrive* observation augmented with six victim-relative features: relative distance, relative bearing, victim speed, attacker speed, relative longitudinal displacement, and relative lateral displacement. These features expose the victim-attacker relation needed for manoeuvre learning while preserving the black-box threat model.

Control is defined by

$$a_t = (\delta_t, u_t) \in [-1,1]^2, \tag{9}$$

where δ_t is steering and u_t is throttle/brake. For value-based victims, a discrete wrapper provides

$$\mathcal{A}_{disc} = \{\delta_i\}_{i=1}^{5} \times \{u_j\}_{j=1}^{5}. \tag{10}$$

Controlled experiment factors The benchmark fixes core environment controls to isolate attacker effects from uncontrolled simulator variation. Base controls include two agents, no respawn, a 1000-step horizon, deterministic scenario seed pools, and configurable traffic density. Spawn presets define attacker placement as an explicit controlled factor. *near*, *med*, and *far* place the attacker approximately 10, 20, and 40 metres behind the victim, respectively. *random* samples uniformly between the near and far offsets.

Reward term	Weight
Indirect victim collision	100.0
Victim off-road event	100.0
Victim mission success	-100.0
Attacker crash	-100.0
Attacker off-road event	-100.0
Following distance	0.3
Closing distance	2.0
Low TTC	2.0
Victim lateral deviation	3.0
Victim deceleration	0.5
Attacker velocity change	-0.5
Per-step cost	-0.01

Table 2: Attacker reward weights used in the proposed configuration. Top: terminal outcome terms. Bottom: Step-level terms.

Algorithm	Key settings
DQN	MLP 259-256-256-25, lr 1×10^{-4}, replay 100k, warm-up 10k, batch 64, $\gamma = 0.99$, target update 1k.
DDQN	DQN profile with Double-Q target selection/ evaluation split.
PPO	Actor/critic MLP 259-256-256, lr 3×10^{-4}, rollout 2048, batch 64, epochs 10, $\gamma = 0.99$, GAE $\lambda = 0.95$, clip 0.2.
SAC	Policy MLP 259-256-256, twin Q MLP (259+2)-256-256, lr 3×10^{-4}, replay 100k, warm-up 10k, batch 256, $\gamma = 0.99$, $\tau = 0.005$.

Table 3: Core victim hyperparameter profile. Here, lr denotes learning rate, GAE denotes generalised advantage estimation, γ is the discount factor, and τ is the SAC soft-update coefficient.

Adversarial wrapper A dedicated `AdversarialEnv` wrapper maps two-agent simulator dynamics into a single-agent attacker-learning interface while stepping the frozen victim internally at each timestep. At each step, the wrapper obtains the frozen victim action, coerces both victim and attacker actions into the *MetaDrive* action space, applies the joint action to the simulator, and returns the attacker observation and reward. During this transition, the wrapper records victim collision, victim off-road, victim arrival, direct attacker-victim collision, TTC, and reward terms under a fixed episode-termination rule.

Attacker reward The attacker reward is computed at each simulator step, and the reported episode reward is the sum of the per-step rewards. The reward combines terminal and step-level signals to encourage victim mission violations without relying on direct attacker-victim collisions or attacker failures. Terminal outcome terms reward indirect victim collisions and victim off-road events, while victim arrival, attacker crashes, and attacker off-road events are treated as attacker failures or penalties. Distance-based credit assignment reduces outcome reward when the attacker is too far from the victim to plausibly contribute to the failure. Step-level shaping terms reward low TTC, closing behaviour, following distance, victim braking, and victim lane deviation. The bottom block of Table 2 also includes a penalty for abrupt attacker motion, measured as the norm of the velocity change between consecutive steps, and a small per-step cost.

Reproducibility controls Each scenario is generated from a controlled seed pool, and each evaluation condition maps episodes deterministically into that pool. This makes the generated road instance, attacker placement procedure, victim policy, attacker policy, and driving setting repeatable for the same condition. The evaluation records per-episode outcomes, aggregate metrics, and run metadata to support auditing, reproduction, and comparison of benchmark runs.

5 Experimental Setup

Figure 3 summarises the experimental pipeline used in the current evaluation. First, victim policies are trained without an attacker to learn the driving task before adversarial interaction is introduced. Second, the trained victim policy is kept fixed while an attacker is trained against it. Finally, evaluation runs cover the selected combinations of policies, attacker spawn presets, traffic densities, and seeds defined by the evaluation matrix.

Victim and attacker training Agents are defined through a policy interface compatible with *Stable-Baselines3* [11]. In the current evaluation, victim policies are trained with DQN [9], DDQN [17], PPO [13], and SAC [6]. Their core hyperparameters are reported in Table 3. Similarly, RL attackers are trained against fixed victim policies through the same policy interface, whereas scripted attackers use fixed heuristics and require no training. The no-op attacker baseline keeps the attacker vehicle present but applies zero steering and zero throttle or braking, providing a passive baseline for attacker comparisons.

The action space follows the benchmark interfaces defined in Section 4. DQN and DDQN use the discrete 25-action wrapper, whereas PPO and SAC use continuous control. In our experiments, both victim training and PPO attacker training use 500,000 simulator steps before evaluation. Each completed evaluation condition is then run for 100 episodes.

Evaluation matrix The highway attacker evaluation fixes the road geometry to highway and the victim to the PPO pol-

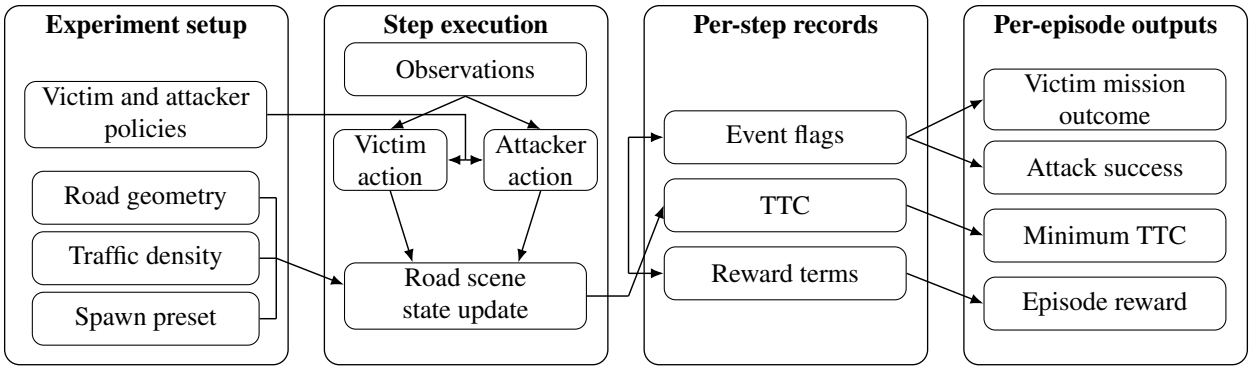


Figure 2: Benchmark design from experiment setup to per-episode outputs.

icy trained on highway. The PPO highway victim used in this comparison is trained with traffic density 0.1. The highway attacker evaluation then considers two traffic settings for the attacker comparison. The primary controlled setting removes traffic and compares the no-op attacker baseline with the PPO attacker trained under the *med* attacker spawn preset, to directly measure attacker influence. A second setting repeats the same comparison with traffic density 0.1 to test whether the attacker effect persists when traffic is present. Both settings use start seed 42 for *MetaDrive* procedural road generation and the same deterministic highway scenario pool for both attacker configurations.

Roundabout is used to study how victim learning changes under a more complex road geometry. The reported roundabout evaluation is limited to driving competence before an attacker is introduced. It varies the victim algorithm across DQN, DDQN, PPO, and SAC. Victim driving competence is reported over the final 100 training episodes of each victim training run.

Evaluation metrics The evaluation metrics follow the outcomes defined in Section 3. Victim driving competence is reported through mission success, off-road rate, and crash rate. The primary comparison is between mission success and attack success: mission success requires the victim to reach its destination without a safety violation, whereas attack success requires a victim off-road event or a victim collision that does not involve direct contact with the attacker. Specifically, ASR is the fraction of episodes in which this attack-success event occurs. To characterise successful attacks beyond their frequency, the evaluation also considers Time-to-failure (TTF) and TTC. TTF measures how many steps pass before an attack succeeds, and is therefore computed only over successful-attack episodes. TTC measures the time to a possible attacker-victim collision under the current relative motion, and is used to interpret close interactions rather than to define attack success. Attacker crash and reward metrics are reported alongside victim outcomes to distinguish victim degradation from attacker failure. To evaluate attacker realism, the evaluation also uses attacker crash rate and attacker velocity change as indicators of whether the attack remains plausible and, indirectly, stealthy.

6 Results

This section reports the conducted experiments used to evaluate victim driving competence, attacker learning, and highway attack effectiveness. The roundabout attacker evaluation remains work in progress; the current roundabout evidence is therefore limited to victim driving competence.

6.1 Experiment 1: Victim Driving Competence in Highway and Roundabout

This experiment evaluates victim driving competence before adversarial evaluation. Table 4 reports mission success, off-road rate, and crash rate over the last 100 training episodes for each completed victim run. On highway, the PPO victim reaches 86.0% success, with 15.0% off-road rate and no crashes, providing a suitable baseline for the attacker evaluation. At the same time, PPO falls to 36.0% success in roundabout, with 56.0% off-road rate and 15.0% crash rate. Notably, DQN and DDQN do not obtain competence in the roundabout road geometry, with 0.0% success rate and 100.0% off-road rate. Finally, SAC provides the strongest roundabout victim, with 57.0% success, 29.0% off-road rate, and 13.0% crash rate. Table 4 summarises these findings. The corresponding victim training success curves are shown in Appendix A.

`Takeaway.` Preliminary results indicate that victim driving competence depends on the interaction between road geometry and policy-learning algorithm.

6.2 Experiment 2: Attacker Learning Dynamics

This experiment checks whether the PPO attacker learns to induce victim failures in the reported highway setting. The

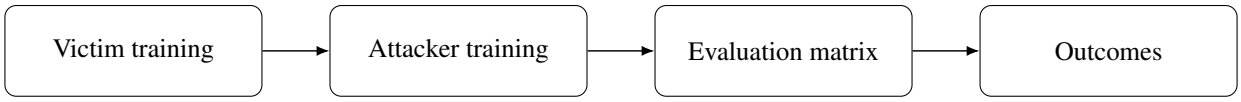


Figure 3: Experimental pipeline from victim training to outcome reporting.

Victim run	Success	Off-road	Crash
PPO highway	86.0%	15.0%	0.0%
DQN roundabout	0.0%	100.0%	0.0%
DDQN roundabout	0.0%	100.0%	0.0%
PPO roundabout	36.0%	56.0%	15.0%
SAC roundabout	57.0%	29.0%	13.0%

Table 4: Victim driving competence over the last 100 training episodes.

attacker is trained against the fixed PPO highway victim under the *med* attacker spawn preset for 500,000 simulator steps. Without traffic, victim off-road failures occur in 49.0% of the final 100 training episodes. The same final window has mean episode reward 25.8 and mean episode length 104.2 steps. With traffic density 0.1, victim off-road failures occur in 51.0% of the final 100 training episodes, while victim crashes remain at 0.0%. The corresponding final window has mean episode reward 82.3 and mean episode length 187.6 steps. These outcomes indicate that the PPO attacker remains able to operate in traffic and induce victim off-road failures. The corresponding attacker training curves are shown in Appendix A.

`Takeaway.` The PPO attacker learns to induce victim off-road failures in the highway setting and remains effective with traffic density 0.1.

6.3 Experiment 3: Highway Attack Effectiveness

This experiment quantifies safety degradation caused by a learned attacker in the highway environment. Table 5 reports outcomes for a PPO victim under the no-op attacker baseline and PPO attacker over 100 recorded episodes per configuration.

Without traffic, the PPO attacker increases ASR from 0.49 to 0.59. This increase is entirely due to victim off-road failures, as victim crashes remain at 0.00 in both attacker configurations. With traffic density 0.1, the same comparison remains positive but weaker, with ASR increasing from 0.42 to 0.46. Nevertheless, the PPO attacker still increases ASR, but the effect is smaller than in the no-traffic setting. This is consistent with a more complex interaction in which traffic adds constraints and occasional attacker failures. At the same time, the traffic setting introduces a small rate of victim crashes involving traffic, which is obviously absent without traffic.

Metric	No-op attacker	PPO attacker
ASR_{crash}	0.00	0.00
$ASR_{offroad}$	0.49	0.59
ASR	0.49	0.59
Mean TTF (steps)	40.51	40.14
Mean episode length	100.49	89.26
ASR_{crash}	0.00	0.01
$ASR_{offroad}$	0.42	0.45
ASR	0.42	0.46
Mean TTF (steps)	85.57	99.57
Mean episode length	203.09	207.30

Table 5: Highway evaluation outcomes for the PPO victim under the *med* attacker spawn preset. Top: without traffic. Bottom: traffic density 0.1.

The corresponding ASR summary and TTF distributions are shown in Appendix A.

`Takeaway.` The PPO attacker increases ASR in both highway traffic settings, while traffic reduces the effect size and introduces a small rate of victim crashes involving traffic.

6.4 Experiment 4: Trade-offs between Effectiveness and Realism

For an adversarial manoeuvre to be security-relevant, it must increase attack effectiveness without relying on frequent attacker crashes or visibly irregular driving. Therefore, this experiment evaluates how attack effectiveness is balanced with attacker realism through crash rate, mean reward, and velocity change for the same highway configurations considered in Experiment 3. Table 6 reports these metrics for the no-op attacker baseline and PPO attacker.

Without traffic, the PPO attacker increases mean reward from 32.02 to 38.88 and does not increase crash rate. However, velocity change rises from 0.08 to 3.55, indicating more abrupt attacker motion than the no-op attacker baseline. With traffic density 0.1, the PPO attacker remains effective, but traffic constrains the manoeuvre. As a consequence, mean reward is lower than the no-op attacker baseline, which is consistent with traffic acting as an obstacle to the attacker. Nevertheless, crash rate remains low at 0.02, and velocity change is lower than in the no-traffic PPO attacker setting. These results indicate that the PPO attacker remains effective under traffic while partially preserving realism and stealthiness.

Metric	No-op attacker	PPO attacker
Crash rate	0.00	0.00
Mean reward	32.02	38.88
Velocity change	0.08	3.55
Crash rate	0.00	0.02
Mean reward	95.21	68.87
Velocity change	0.08	1.44

Table 6: Attacker-side realism indicators for the PPO victim under the *med* attacker spawn preset. Top: without traffic. Bottom: traffic density 0.1.

Takeaway. Traffic reduces PPO attacker effectiveness, but the learned policy still maintains a favourable trade-off between attack effectiveness and realism.

7 Discussion

The benchmark provides a controlled setting for studying how LIDAR-based driving policies respond to adversarial manoeuvres performed through ordinary vehicle control.

The results suggest that such policies can be susceptible to physically valid attacker behaviour, but they also show that attack evaluation is easy to misinterpret without a structured protocol. In particular, victim competence, attacker effectiveness, and attacker realism must be separated before an attack outcome can be interpreted. This separation is important because an observed failure can otherwise reflect weak victim driving, an unrealistic attacker, or a genuine adversarial effect.

The roundabout results illustrate this point. Before adversarial evaluation is meaningful in that road geometry, the victim policy must first demonstrate sufficient driving competence. The current results indicate that this competence depends on the policy-learning algorithm, which motivates further training, diagnosis of the DQN and DDQN failures, and evaluation of additional victim policies.

The current benchmark models a single attacker vehicle and uses a LIDAR-based observation interface. The same protocol can be extended along the controlled factors already exposed by the benchmark, including road geometry, traffic density, attacker spawn preset, and procedural seed. Future work should extend the training budget for both victims and attackers, diagnose the DQN and DDQN roundabout failures, and evaluate further victim–attacker algorithm combinations. The evaluation matrix should also be expanded to cover additional attacker spawn presets, traffic densities, and road geometries.

8 Ethical Boundary

This work is intended for defensive security evaluation. All experiments are simulation-only and are used to assess robustness to adversarial manoeuvre attacks under a constrained, physical-interaction-only attacker model. The paper does not provide real-world deployment procedures for adversarial behaviour, and the benchmark is intended for controlled simulation and robustness assessment only.

Because the paper studies safety-critical failure induction in autonomous-driving settings, ethical and legal risks were considered during the design of the evaluation protocol. To mitigate these risks, the study excludes attacks outside physical vehicle manoeuvres (e.g., sensor spoofing, communication tampering, and code tampering) and reports results only in simulation.

9 Conclusion

This paper presents a reproducible benchmark for adversarial manoeuvre attacks in Autonomous Driving. It instantiates the attacker-victim interaction in *MetaDrive*, fixes the attacker capability and success criteria, and reports victim outcomes together with attacker-side realism indicators.

The current evaluation shows that a PPO attacker can increase highway ASR against a PPO victim while remaining within ordinary vehicle control. It also shows that this result must be interpreted together with attacker crash rate, velocity change, and the driving competence of the victim policy.

The victim results further show that driving competence depends on the road geometry and policy-learning algorithm. This makes victim validation a necessary part of adversarial manoeuvre evaluation, especially before extending the attack study to roundabout settings.

The benchmark is intentionally scoped to the conducted experiments. Its main contribution is to make adversarial manoeuvre evaluation repeatable, explicit, and extensible across road geometries, traffic settings, attacker spawn presets, and policy combinations.

These results support further systematic testing of adversarial manoeuvre attacks against Autonomous Driving policies.

References

[1] Karl Johan Åström. Optimal control of markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 1965.

[2] Yulong Cao, S. Hrushikesh Bhupathiraju, Pirouz Naghavi, Takeshi Sugawara, Z. Morley Mao, and Sara Rampazzi. You can't see me: Physical removal attacks on LiDAR-based autonomous vehicles driving frameworks. In *32nd USENIX Security Symposium (USENIX Security 23)*, 2023.

[3] Yulong Cao, Ningfei Wang, Chaowei Xiao, Dawei Yang, Jin Fang, Ruigang Yang, Qi Alfred Chen, Mingyan Liu, and Bo Li. Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. In *2021 IEEE Symposium on Security and Privacy (SP)*, 2021.

[4] Jiarun Dai, Bufan Gao, Mingyuan Luo, Zongan Huang, Zhongrui Li, Yuan Zhang, and Min Yang. Sctrans: Constructing a large public scenario dataset for simulation testing of autonomous driving systems. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024.

[5] Ambra Demontis, Srishti Gupta, Maura Pintor, Luca Demetrio, Kathrin Grosse, Hsiao-Ying Lin, Chengfang Fang, Battista Biggio, and Fabio Roli. Security of deep reinforcement learning for autonomous driving: A survey, 2025.

[6] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[7] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 1998.

[8] Aishan Liu, Jiakai Wang, Tianyuan Zhang, Hainan Li, Jiangfan Liu, Siyuan Liang, Yilong Ren, Xianglong Liu, and Dacheng Tao. Metadv: A unified and interactive adversarial testing platform for autonomous driving. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025.

[9] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 2015.

[10] Tong Nie, Yuewen Mei, Yihong Tang, Junlin He, Jie Sun, Haotian Shi, Wei Ma, and Jian Sun. Steerable adversarial scenario generation through test-time preference alignment. In *The Fourteenth International Conference on Learning Representations*, 2026.

[11] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 2021.

[12] Takami Sato, Junjie Shen, Ningfei Wang, Yunhan Jia, Xue Lin, and Qi Alfred Chen. Dirty road can attack: Security of deep learning based automated lane centering under Physical-World attack. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

[13] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[14] Aizaz Sharif and Dusica Marijan. Adversarial deep reinforcement learning for improving the robustness of multi-agent autonomous driving policies. In *2022 29th Asia-Pacific Software Engineering Conference (APSEC)*, 2022.

[15] Ruoyu Song, Muslum Ozgur Ozmen, Hyungsub Kim, Raymond Muller, Z. Berkay Celik, and Antonio Bianchi. Discovering adversarial driving maneuvers against autonomous vehicles. In *Proceedings of the 32nd USENIX Security Symposium*, 2023.

[16] Benjamin Stoler, Ingrid Navarro, Jonathan Francis, and Jean Oh. Seal: Towards safe autonomous driving via skill-enabled adversary learning for closed-loop scenario generation. *IEEE Robotics and Automation Letters*, 2025.

[17] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

[18] Chejian Xu, Wenhao Ding, Weijie Lyu, ZUXIN LIU, Shuai Wang, Yihan He, Hanjiang Hu, DING ZHAO, and Bo Li. Safebench: A benchmarking platform for safety evaluation of autonomous vehicles. In *Advances in Neural Information Processing Systems*, 2022.

[19] Jiawei Zhang, Chejian Xu, and Bo Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[20] Linrui Zhang, Zhenghao Peng, Quanyi Li, and Bolei Zhou. CAT: Closed-loop adversarial training for safe end-to-end driving. In *Conference on Robot Learning (CoRL)*, 2023. MetaDrive-based closed-loop adversarial scenario generation.

A Additional Plots

This appendix reports supporting plots for the conducted experiments.

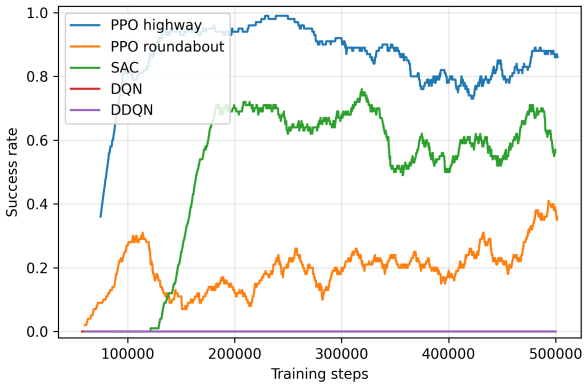


Figure 4: Victim training success rate over training steps. PPO is shown for both highway and roundabout; SAC, DQN, and DDQN are shown for roundabout.

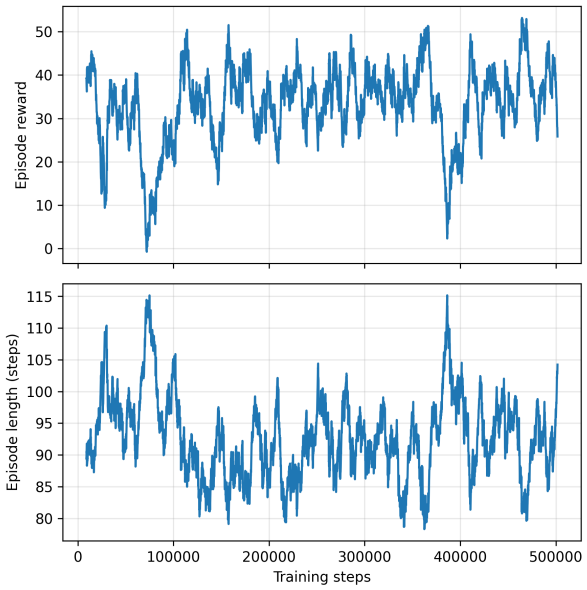


Figure 5: PPO attacker training curves in the highway setting under the *med* attacker spawn preset.

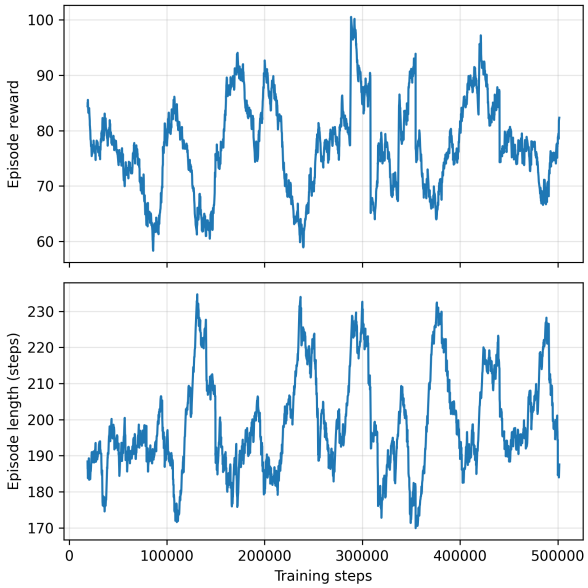


Figure 6: PPO attacker training curves in the highway setting with traffic density 0.1 under the *med* attacker spawn preset.

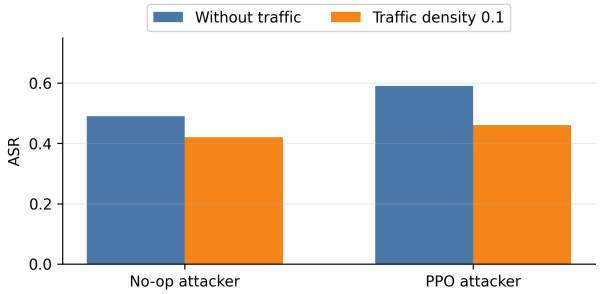


Figure 7: ASR summary for the highway evaluation under the *med* attacker spawn preset.

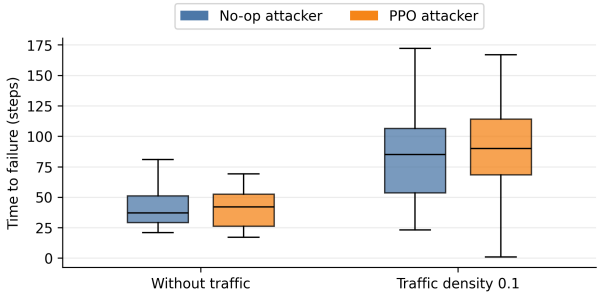


Figure 8: TTF distributions for the highway evaluation under the *med* attacker spawn preset.

B Artifact Appendix

B.1 Abstract

This artifact contains the source code, configuration files, pre-trained models, and pre-computed outputs needed to reproduce the experimental results reported in the paper.

B.2 Description & Requirements

The artifact consists of approximately 3,000 lines of Python source code organised under `src/`, experiment configuration files under `configs/`, and pre-trained victim and attacker policy checkpoints under `outputs/paper/models/`. A single CLI command (`python main.py --config configs/paper.jsonc`) executes the complete pipeline: victim training, attacker training, and evaluation with metric computation. Post-processing utilities under `scripts/` generate result tables and figures from the evaluation outputs.

B.2.1 Security, privacy, and ethical concerns

The artifact runs simulation-only experiments in the MetaDrive driving simulator. No network connections are required during execution, no real vehicles or physical systems are involved, and no destructive operations are performed on the host machine. The trained attacker policies are reinforcement-learning models for a simulated environment and are not deployable on real vehicles.

B.2.2 How to access

The artifact, including the source code, configuration files, and pre-trained models, is archived on Zenodo at https://doi.org/10.5281/zenodo.20530026.

B.2.3 Badges

The artifact has been submitted for the Available, Functional, and Results Reproduced badges.

B.2.4 Hardware dependencies

- **CPU**: Modern x86-64 or ARM64 processor (8+ cores recommended).
- **Memory**: 8 GB RAM minimum, 16 GB recommended.
- **Disk**: 20 GB free space for the virtual environment, trained models, and evaluation outputs.
- **GPU**: Not required.
- **Display**: Not required (rendering is disabled by default).

B.2.5 Software dependencies

- Python 3.11.
- Pinned dependencies in `requirements.txt`.
- **Ubuntu only:** OpenGL runtime package `libgl1`.

Tested on Ubuntu 22.04 (x86-64) and macOS 26 (ARM64). On macOS, `--device cpu` is the recommended configuration.

B.2.6 Benchmarks

None.

B.3 Set-up

B.3.1 Installation

1. Download the artifact archive and extract it.
2. Create a virtual environment and install dependencies:

```
sudo apt install -y libgl1  # Ubuntu only
python3.11 -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
```

B.3.2 Basic Test

Run the smoke-test configuration, which trains a short PPO victim and evaluates it with a no-op attacker:

```
python main.py \
  --config configs/test.jsonc \
  --experiment-name test --device cpu
```

Expected wall time is approximately 2 minutes. The command should create `outputs/test/experiment.json` and metric files under `outputs/test/eval/`.

B.4 Evaluation workflow

B.4.1 Major Claims

(C1): Victim robustness is algorithm-dependent: on the roundabout scenario, SAC achieves 57% success (mean reward 191.5), PPO 36% (177.5), while DQN and DDQN achieve 0% success with 100% off-road rate. On the highway, PPO achieves 86% success (mean reward 238.9). See Experiment 1 in Section 6.1, Table 4, and Figure 4.

(C2): The learned PPO attacker converges to positive mean reward (≈25.8) on the highway environment with a fixed spawn geometry over 500,000 training steps. See Experiment 2 in Section 6.2 and Figure 5.

(C3): In the highway evaluation with the *med* attacker spawn preset, a learned PPO attacker raises ASR from 0.49 to 0.59 without traffic, and from 0.42 to 0.46 with traffic density 0.1. See Experiment 3 in Section 6.3, Table 5, and Figures 7–8.

(C4): The highway evaluation reports attacker crash rate, mean reward, and velocity change to interpret the trade-off between attack effectiveness and attacker realism. With traffic density 0.1, the PPO attacker shows lower velocity change and low crash rate, indicating less abrupt

manoeuvres in traffic while remaining effective. See Experiment 4 in Section 6.4 and Table 6.

B.4.2 Experiments

All paper experiments are generated by one pipeline run, followed by dedicated scripts under `scripts/` for E1–E4.

(E0): Pipeline execution (for all experiments) [10 human-min + 12–18 compute-hr + 20 GB disk]: Runs victim training, attacker training, and evaluation in a single command, producing all outputs required by E1–E4.
Preparation: Complete the Set-up section and ensure 20 GB free disk.
Execution:
```
source .venv/bin/activate
python main.py \
  --config configs/paper.jsonc \
  --experiment-name paper \
  --num-jobs 1 --device cpu
```
With `-num-jobs 12`, wall time is approximately 8–12 hours on a 16-core machine.
Results: Outputs appear under `outputs/paper/`, including models and evaluation results.

(E1): Victim learning quality [5 human-min + 1 compute-min] (validates C1):
Preparation: Verify that victim training directories exist under `models/victims/`, one per algorithm and scenario (e.g., `ppo_highway`, `sac_roundabout`), each containing a `training_episodes.csv` file.
Execution:
```
python scripts/make_victim_training_table.py
```
Results: The script prints success rate, off-road rate, crash rate, and mean reward per victim algorithm over the last 100 training episodes. The expected values match Section 6.1: on the roundabout, SAC success ≈57%, PPO ≈36%, DQN/DDQN ≈0% with 100% off-road; on the highway, PPO success ≈86%.

(E2): Attacker learning dynamics [5 human-min + 1 compute-min] (validates C2, Figure 5):
Preparation: Verify that attacker training directories exist under `models/attackers/`, named by attacker algorithm, victim algorithm, scenario, and spawn geometry (e.g., `ppo_vs_ppo_highway_med`), each containing a `training_episodes.csv` file. Victim training directories from E1 must also be present.
Execution:
```
python scripts/\
plot_victim_training_curves.py
python scripts/\
plot_attacker_training_curves.py
```
Results: The first script produces Figure 4 (victim training success curve) and the second produces Figures 5–6 (attacker reward and episode length curves), both saved under `outputs/plots/`. The highway attacker curve should show convergence to positive mean reward (≈25.8 over the last 100 training episodes).

(E3): Highway attack effectiveness [5 human-min + 1 compute-min] (validates C3, Table 5, Figures 7–8):
Preparation: Verify that evaluation directories exist under `eval/`, named by scenario, victim, attacker, spawn geometry, and seed, each containing `metrics.json` and `episodes.csv`.
Execution:
```
python scripts/make_eval_table.py
python scripts/plot_attack_success_rate.py
python scripts/\
plot_time_to_failure_distribution.py
```
Results: The first script prints Table 5, where ASR increases from 0.49 to 0.59 without traffic and from 0.42 to 0.46 with traffic density 0.1. The plotting scripts produce Figure 7, which compares ASR for the no-op and PPO attackers, and Figure 8, which shows the TTF distributions for the same two traffic settings.

(E4): Effectiveness–realism trade-off [5 human-min + 1 compute-min] (validates C4, Table 6):
Preparation: The same evaluation directories from E3 are used. Verify that the evaluation outputs contain the metrics used for attacker crash rate, mean reward, and velocity change.
Execution:
```
python scripts/make_realism_table.py
```
Results: The script prints Table 6 (attacker crash rate, mean reward, and velocity change). The expected Table 6 output without traffic is crash rate 0.00 for both attackers, mean reward 32.02 and 38.88, and velocity change 0.08 and 3.55. With traffic density 0.1, the expected output is crash rate 0.00 and 0.02, mean reward 95.21 and 68.87, and velocity change 0.08 and 1.44.

B.5 Notes on Reusability

The benchmark is designed around three main extension points:

- **Scenarios:** Add a JSON file to `configs/scenarios/` specifying the MetaDrive environment class and configuration, then reference the scenario name in `configs/paper.jsonc`.
- **Victim/attacker algorithms:** Add policy classes that inherit from Stable-Baselines3 algorithms and register them in the agent registry. Discrete and continuous action-space mapping is handled automatically.
- **Reward design:** Add or modify presets in `src/reward/config.py`. The reward engine evaluates a configurable weighted sum of 16 components.

B.6 Version

Based on the LaTeX template for Artifact Evaluation V20260101. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/vehiclesec2026/.