PLANNER-BREAKER: FINDING INFORMATIVE FAILURES OF DRIVING PLANNERS WITH REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Closed-loop driving benchmarks assess nominal performance among logged or rule-based traffic, but reveal little about a planner's weaknesses. Adversarial methods address this by generating safety-critical situations, but an identified collision is only informative if the planner could have avoided it. We introduce Planner-Breaker, which uses self-play reinforcement learning at scale to train traffic that produces collisions of a chosen form against a given black-box planner. Since this reward does not need to be differentiable, we instantiate it with a counterfactual criterion that tests whether braking in time would have avoided a collision, and whether the danger was foreseeable early enough to react. Evaluated against five rule-based and learned planners, Planner-Breaker produces about three times more avoidable collisions than gradient-based adversaries, which collide about as often or less. A cross-evaluation shows that each adversary exploits weaknesses specific to the planner it was trained on. Our adversaries can be used without retraining as a traffic model in real-world scenes, where they produce more informative collisions than log replay and rule-based traffic. The same method optimizes other failure criteria, such as nuPlan's at-fault rule, allowing practitioners to search their own planner for the failures they care about.

1 INTRODUCTION

Recent closed-loop planners report strong performance on established benchmarks, with methods such as PDM (Dauner et al., 2023) and GIGAFLOW (Cusumano-Towner et al., 2025) reaching near-saturated scores. These scores are obtained against traffic that either reacts to the planner through cooperative rule-based models or replays logged trajectories (Karnchanachari et al., 2024; Charraut et al., 2025), where dangerous interactions are rare. Performance is then averaged over many scenarios, so the few safety-critical interactions contribute little to the final score. A strong benchmark number therefore says little about how a given planner behaves in dangerous interactions.

To expose this behavior, adversarial scenario generation produces scenes and traffic behavior under which a planner fails (Ransiek et al., 2025). It serves three purposes: generating training data for more robust planners (Hanselmann et al., 2022; Zhang et al., 2023; Stoler et al., 2025; Nie et al., 2026; Liu et al., 2025), building challenging scenarios to evaluate planners (Feng et al., 2023; Ransiek et al., 2024), and targeting a given planner to analyze or improve it (Koren et al., 2018; Wachi, 2019; Rempe et al., 2022). We focus on the third, where failures are only useful if they are informative, that is, if they reveal something the planner could have done differently. An adversary that rams the vehicle driven by the planner, the *target*, from behind causes a collision, but tells us nothing about the planner.

Gradient-based methods optimize for collisions and, at best, separate informative ones outside the objective through a proxy (Hanselmann et al., 2022; Rempe et al., 2022; Yin et al., 2024). A standard proxy is the *at-fault* criterion of nuPlan (Karnchanachari et al., 2024), a geometric rule on who hit whom at the time of impact, which does not consider whether the collision could have been avoided. We instead call a collision *avoidable* when the target could have avoided it by braking earlier, and *informative* when, in addition, the danger was foreseeable early enough to do so. We do not ask who is to blame for the collision, but whether the planner did what it could to avoid it.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Because reinforcement learning (RL, Sutton & Barto, 2018) only requires a scalar reward, this criterion can directly serve as the training objective, without the differentiable proxies that gradient-based methods require. We propose *Planner-Breaker*, which builds on self-play RL for driving (Cusumano-Towner et al., 2025; Cornelisse et al., 2025b; 2026), and turns it against a fixed black-box planner: a single shared policy controls all surrounding traffic. Episodes start at random locations across entire maps, with no scenario designed by hand; each adversary has its own driving goals, is rewarded for driving like ordinary traffic, and learns by itself where and when to attack. Planner-Breaker thereby discovers strategies that expose the weaknesses of the planner it is trained against. The result is a policy rather than a set of scenarios: once trained, it can be rolled out on new initial scenes without further optimization.

We evaluate Planner-Breaker against five planners. Compared to gradient-based adversaries (Hanselmann et al., 2022; Yin et al., 2024), it causes a similar or higher number of collisions but, under our definitions, about three times as many avoidable ones and more than four times as many informative ones. In a cross-evaluation, each adversary is most effective against the planner it was trained on, showing that it exploits weaknesses specific to that planner. Adversaries rewarded for any collision show no such specificity: their collisions are mostly unavoidable, and almost equally frequent against every planner. Planner-Breaker can also be trained to optimize nuPlan's at-fault rule (Caesar et al., 2021; Karnchanachari et al., 2024) instead of our criterion. Finally, deployed without retraining on real-world scenes (Ettinger et al., 2021), it produces more informative collisions than both log replay and rule-based traffic. We release our code, together with an adversary trained against all five planners that can be tried on other planners before training a dedicated one.

Our contributions are as follows:

1. We propose Planner-Breaker, which turns self-play RL at scale against a fixed black-box planner. The result is a reusable adversarial policy that can be rolled out on new initial scenes without further optimization. Since its reward only has to be computed, not differentiated, it can optimize failure criteria that gradient-based adversaries cannot, such as counterfactually avoidable collisions.
2. We propose a counterfactual avoidability test to evaluate whether collisions induced by adversarial drivers are informative. Applied to gradient-based adversaries and to adversaries rewarded for any collision, it shows that most of their collisions are unavoidable. Among avoidable collisions, a foreseeability test further separates those whose danger was visible early enough to react.
3. Against five planners, we show that each adversary learns the weaknesses of the planner it is trained against, and that it produces more avoidable and informative collisions than gradient-based methods, at a similar or higher collision rate. Deployed without retraining on real-world scenes, it also produces more informative collisions than log replay and rule-based traffic.

2 Related Work

Adversarial scenario generation. Simulated traffic can be made adversarial to expose the failures of a driving system (see Ransiek et al. (2025) for a review), and a family of methods uses such scenarios as training data for a more robust driving policy (Zhang et al., 2023; Stoler et al., 2025; Nie et al., 2026; Liu et al., 2025). Another family optimizes a scene against a given planner: KING perturbs the trajectories of adversarial agents by gradient descent through a kinematic bicycle model, while the planner reacts in closed loop (Hanselmann et al., 2022), and ReGentS makes this optimization more stable on real-world scenes (Yin et al., 2024). STRIVE jointly optimizes all agents of a scene in the latent space of a learned traffic model, and represents the planner by a differentiable proxy within this model (Rempe et al., 2022). These methods optimize each scene separately and require a differentiable model of the dynamics, the simulator or the planner. Their objective is a distance between the adversaries and the target. KING and ReGentS average this distance over the whole scenario, which penalizes strategies that first move away from the target. Planner-Breaker instead trains one policy per planner with RL, which only needs a scalar reward. The adversarial reward depends on the target's collision, not on the path that leads to it. New scenarios are generated by rolling out this policy.

Learned adversaries. Adaptive stress testing (AST) uses RL or tree search to find the most likely failure trajectory of a black-box system in a given scenario (Lee et al., 2020; Koren et al., 2018). Other methods use RL without driving the adversary directly: Learning to Collide sets where and when an obstacle appears in hand-designed scenarios (Ding et al., 2020), GOOSE sets the whole trajectory of an adversarial vehicle in pre-crash scenarios from a lane-keeping regulation (Ransiek et al., 2024), and D2RL sets the probability that a background vehicle performs a predefined adversarial manoeuvre, to estimate crash rates without bias (Feng et al., 2023). Closer to our setting, RL agents drive adversarial vehicles step by step against a fixed system: FailMaker-AdvRL trains one or two vehicles with MADDPG, adding a personal driving reward to the adversarial reward (Wachi, 2019), and Cai et al. (2025) train a series of single adversaries with TD3 in two hand-designed scenarios. These methods control one or a few agents, often in hand-designed scenarios. In Planner-Breaker, one shared policy drives every surrounding vehicle, and episodes start at random locations across entire maps.

Informative failures. Collision-seeking adversaries often find collisions that say little about the system under test: in AST, most returned collisions were blamed on the pedestrian, and restricting failures to those the system is responsible for was left to future work (Koren et al., 2018). FailMaker-AdvRL (Wachi, 2019) and Cai et al. (2025) report the same issue with adversaries that intentionally collide with the tested vehicle. Existing methods handle this outside an objective that still favors any collision: KING and STRIVE keep generated scenarios in which a privileged rule-based expert or a solution optimization finds a trajectory that avoids the collision (Hanselmann et al., 2022; Rempe et al., 2022), ReGentS excludes adversaries located behind the ego vehicle before optimization, since rear-end collisions often cannot be avoided (Yin et al., 2024), and Cai et al. (2025) add a term for RSS-critical frames to a reward paid on every collision. Benchmarks apply similar rules at evaluation: the at-fault criterion of nuPlan penalizes a planner only when it could be responsible for the collision, based on the configuration at impact (Karnchanachari et al., 2024). Since these criteria assign a class to each collision, they provide no gradient, and gradient-based methods can only apply them outside the objective. Planner-Breaker instead rewards each collision according to whether the planner could have avoided it: collisions that braking earlier would have avoided are rewarded, more so when the danger was foreseeable early enough to react, in the spirit of the proper response of RSS (Shalev-Shwartz et al., 2018), and unavoidable collisions are penalized (Section 3.3).

Self-play at scale. Self-play RL has produced strong agents in games such as Go, Dota 2 and StarCraft II (Silver et al., 2018; OpenAI et al., 2019; Vinyals et al., 2019). In driving, batched simulators make self-play at scale practical (Kazemkhani et al., 2024; Cornelisse et al., 2025a). Recent driving policies are trained through self-play in such simulators (Cusumano-Towner et al., 2025; Cornelisse et al., 2025b; 2026). In adversarial RL, an adversary is trained against another agent, usually to make it more robust: in RARL, an adversary applies disturbance forces to a protagonist in a zero-sum game (Pinto et al., 2017), and training against a population of adversaries further improves robustness (Vinitsky et al., 2020). Planner-Breaker brings self-play at scale to adversarial driving, with adversaries that share one policy and are trained against a planner that does not learn.

3 PLANNER-BREAKER

3.1 PROBLEM SETUP

We build on the multi-agent setting of GIGAFLOW (Cusumano-Towner et al., 2025): N vehicles are placed on a static map W, each with a physical state x_t^i (position, heading, velocity, dimensions), giving the simulator state $s_t = (W, x_t^0, \ldots, x_t^{N-1})$. Vehicle 0 is the *target*, controlled by a planner f that we treat as a fixed black box, $x_{t+1}^0 = f(s_t)$. All other vehicles are *adversaries*, controlled by a single policy π with parameters shared across them (Yu et al., 2022): adversary i receives a local observation $o_t^i = \mathcal{O}(s_t, i)$, takes an action $a_t^i \sim \pi(o_t^i)$, and receives a reward r_t^i. The observation function $\mathcal{O}$ can be chosen freely, augmented with an observation that identifies the target among the observed vehicles. Our observation function closely follows GIGAFLOW and is detailed in Section A.1.

Since f is fixed, it is part of the environment dynamics, and the adversaries play a partially observed stochastic game (Littman, 1994; Hansen et al., 2004) among themselves. We train

π with PPO (Schulman et al., 2017) to maximize each adversary's expected discounted return $J(\pi) = \mathbb{E}\big[\sum_t \gamma^t r_t^i\big]$, with r_t^i defined in Section 3.2.

3.2 Training the adversaries

Each adversary's reward combines a term shared by all adversaries, which depends on the target's collision, and an individual driving term:

$$r_t^i = r_{\text{adv}}(c_t) + r_{\text{drive}}^i(t).$$

Here $c_t \in \mathcal{C}$ is the class of the target's collision at step t under a chosen criterion, where $\mathcal{C}$ includes a *no collision* class, and r_{adv} maps each class to a reward, with $r_{\text{adv}}(\text{no collision}) = 0$. Our criterion is introduced in Section 3.3; Planner-Breaker can also be trained on the nuPlan at-fault rule (Caesar et al., 2021; Karnchanachari et al., 2024), with $\mathcal{C} = \{\text{at fault}, \text{not at fault}, \text{no collision}\}$ (Section 4.4). Because $r_{\text{adv}}(c_t)$ is shared, an adversary is rewarded for a target collision even when another adversary hits the target, which encourages cooperation. The driving term is inspired by GIGAFLOW; it rewards reaching goals and penalizes collisions, leaving the road, running red lights, and uncomfortable driving (Section A.2). It creates an incentive for adversarial agents to behave like nominal drivers for as long as possible; they drive towards their goals when they are not attacking the target.

Post-collision continuation. A collision or leaving the road terminates an agent's episode on top of the driving reward's penalty. Since the agent forfeits all future rewards on termination, termination is heavily penalized. While this keeps adversaries from colliding with each other, it also discourages them from hitting the target. Moreover, the problem is not fully cooperative, upon termination, the hitter forfeits its future driving reward while the other adversaries keep theirs and also receive $r_{\text{adv}}(c_t)$. Each adversary would be better off if another adversary achieved the hit (Section 4.4). To encourage collisions, we cancel both penalties upon collision with the target: the successful adversary receives no collision penalty and its episode does not terminate. To fix this failure mode, we remove the target from the scene after its collision; the adversaries continue with the driving reward alone for a fixed duration (Section A.3). This continuation also gives the adversaries driving experience after each attack.

Target-focused episodes. As the maps are large, randomly scattering agents across them tends to leave agents far from the target. To use experience efficiently, we place one adversary near the target, on a route that meets the target's, and the others around them. This concentrates the training data on interactions with the target, and even random exploration often leads to collisions with it (Section A.3). For the same reason, the episode ends early when no adversary has been near the target for a short time. Although episodes end early, the adversaries still gain driving experience from continuation.

3.3 Avoidable and Foreseeable collisions

To find collisions that reveal weaknesses of the target, we design a collision criterion for the adversaries to optimize. Some collisions could not have been avoided, and some leave no time to react. The criterion classifies each target collision with two tests, avoidability and foreseeability, and the adversaries are rewarded according to the result.

Avoidability. We test avoidability counterfactually, by asking whether braking would have avoided the collision. Reacting to a danger takes a response time ρ plus the time the target needs to stop, which we call $\tau_{\max}$. Since braking early enough avoids any collision, we only consider braking within $\tau_{\max}$ before the collision time t_c. For each τ from one simulation step $\mathrm{d}t$ to $\tau_{\max}$, we rewind the scene to $t_c - \tau$ and replay it with one change. The target brakes at its maximum deceleration along its recorded path, while the other vehicles replay their recorded motion and continue at constant velocity after t_c. Braking fails if the target still hits the adversary involved in the collision, even after it has stopped. It also fails if it leads the target into another vehicle, but only when the target is at fault under the nuPlan rule. A follower hitting the braking target is ignored, since a real follower would brake too. The smallest τ for which braking succeeds, $\tau^\star$, marks the last moment $t_c - \tau^\star$ at which the target could have braked. If no τ succeeds, the collision is *unavoidable* by braking.

Foreseeability. A collision can be avoidable by braking and still give the target no warning, if the danger appears only after its last chance to react. We test foreseeability only for avoidable collisions, by looking at the recorded scene one response time ρ before the last braking moment $t_c - \tau^\star$. The situation is dangerous if the target and the adversary involved in the collision would collide within $\tau_{\max}$, with both projected at constant velocity or the target projected along its route. It is also dangerous if the adversary is inside a lateral safety margin around the target, which grows with the adversary's lateral speed towards it, in the spirit of RSS (Shalev-Shwartz et al., 2018). If the situation was dangerous, the collision is *foreseeable*, since the danger was visible early enough to react. Otherwise, it is *unforeseeable*.

In our setup, $\mathcal{C} = \{\text{foreseeable}, \text{unforeseeable}, \text{unavoidable}, \text{no collision}\}$, and r_{adv} gives the largest reward to foreseeable collisions, a smaller one to unforeseeable collisions, and a small penalty to unavoidable ones (values and rationale in Section A.4).

4 Experiments

4.1 Experimental setup

Experiments run in PufferDrive (Cornelisse et al., 2025a), a high-throughput multi-agent driving simulator built on `PufferLib` (Suarez, 2025). We follow the GIGAFLOW training setting and train on the same eight CARLA maps (Dosovitskiy et al., 2017), processed with 123D (Dauner et al., 2026), using its neural architecture and discrete action space (Appendix B). We train one shared adversarial policy per target planner for 20B steps, keeping the target fixed throughout. On average, a 20B run takes 1 h 55 min on 8 RTX PRO 6000 GPUs, about 2.9M steps per second.

We consider five target planners, which differ in how they perceive and forecast other vehicles. Line-IDM is an intentionally weak implementation of the Intelligent Driver Model (IDM, Treiber et al., 2000) that only considers the first agent on a straight line in front of it to adapt its speed. IDM considers the first vehicle on its intended path instead, and thus can drive safely in turns. PDM is a reimplementation of PDM-Closed (Dauner et al., 2023), similar to the one in V-Max (Charraut et al., 2025). It scores multiple IDM trajectory candidates while forecasting surrounding vehicles at constant velocity. SP and SP-Cond are two policies trained through self-play for 100B steps in PufferDrive. SP is trained without reward conditioning, similarly to Cornelisse et al. (2025b), and SP-Cond adds the reward conditioning and partner blindness of GIGAFLOW. All targets brake at up to 5 m/s^2, the deceleration used by the avoidability test.

We evaluate on the eight training maps, with 10,000 episodes per run. As in training, one to seven adversaries are initialized around the target, but farther from it, and an episode lasts up to 256 s. The protocol is detailed in Appendix B. We report the share of episodes in which the target collides, split into the classes of Section 3.3. We train 3 adversaries per target with different seeds and evaluate each with 3 evaluation seeds. We report the mean over the 9 runs, and the standard deviation over the 3 training seeds, after averaging the evaluation seeds of each.

We compare with KING (Hanselmann et al., 2022) and ReGentS (Yin et al., 2024), which perturb the trajectories of adversaries in a reference scene by gradient descent through a differentiable kinematic bicycle model (Appendix B). Since they optimize whole trajectories, longer horizons make their optimization harder, and in our tuning their results degraded beyond 100 steps. We therefore optimize them over 100 steps (10 s). Our adversaries have no time budget. Their median collision time is 10.5 s, but 5% of their collisions take more than 89 s.

As a first reference, we place the target among IDM vehicles, the standard traffic model of driving simulators. To compare with learned traffic that is not adversarial, we use the SP policy, trained with the same driving reward but without the adversarial reward. To test whether untargeted perturbations produce informative collisions, we use vehicles taking uniformly random actions. KING, ReGentS and these traffics are not trained against the target, so their standard deviation is over the 3 evaluation seeds.

4.2 FINDING INFORMATIVE FAILURES

Ordinary traffic rarely reveals failures of the targets (Table 1). Except for the intentionally weak Line-IDM, they have a foreseeable collision in at most 1.2% of episodes under IDM or SP traffic. Random traffic makes collisions frequent, but 61 to 93% of them are unavoidable. PDM, which brakes at its maximum deceleration whenever it forecasts a collision within its stopping time, has an avoidable collision in at most 1.6% of episodes under these three traffics.

KING, ReGentS and Planner-Breaker all make the targets collide in most episodes, but not in the same way. Most collisions caused by KING and ReGentS could not have been avoided by the target, while most of ours could (18 to 36% avoidable, against 75 to 94%). Except on PDM, this share is close to that of random traffic, so optimizing for collisions raises their number but not the share that the target could have avoided. At a similar collision rate, Planner-Breaker thus causes 2.7 to 3.8 times more avoidable collisions than KING, and 4.4 to 5.4 times more foreseeable ones (Table 12).

Table 1: **Target collisions under each traffic.** Each bar is 100% of episodes. Its coloured parts are the episodes with a foreseeable, unforeseeable or unavoidable target collision. SP and SP-Cond are self-play RL policies, used as targets and, for SP, as traffic (Section 4.1). Exact rates with standard deviations are given in Table 12.

Traffic \ Target	Line-IDM	IDM	PDM	SP	SP-Cond
IDM					
SP					
Random					
KING					
ReGentS					
Planner-Breaker					

4.3 SPECIALIZATION

Each adversary learns to exploit weaknesses specific to its training target. Each target has the highest rate of foreseeable collisions with the adversary trained on it, and each adversary reaches its highest rate on its training target (Table 2). On the rule-based targets, this adversary leads all others by 17 to 35 points. Even the two learned targets, whose collision rates differ by at most 1.2 points under every traffic of Table 1, are told apart by their adversaries. Each reaches a higher rate on its training target than on the other one, by 5.2 and 6.2 points, on each of its 3 training seeds. The foreseeable collisions of an adversary thus depend on the planner it faces, so they reveal properties of that planner and not only of the traffic. Section 4.4 shows that this specialization disappears when adversaries are rewarded for any collision.

Table 2: **Specialization.** Each cell describes the episodes of the target (column) when all other vehicles are driven by the adversarial policy trained against the planner of the row. The bar spans 100% of episodes, coloured as in Table 1, and the number is the percentage of episodes with a foreseeable target collision. Mixed is trained against all five targets. Means over 3 training $\times$ 3 evaluation seeds.

Adv. \ Target	Line-IDM	IDM	PDM	SP	SP-Cond
Line-IDM	**76.5**	19.6	10.0	36.8	42.9
IDM	35.5	**45.0**	24.7	30.6	30.4
PDM	41.9	22.1	**53.4**	39.8	40.6
SP	29.6	15.3	20.0	**65.1**	59.9
SP-Cond	21.7	10.7	12.1	60.6	**66.8**
Mixed	42.0	28.0	32.6	55.6	54.6

A single adversary trained on all five targets does not replace the adversaries dedicated to each. This adversary (Mixed) has the highest average rate over the five targets (42.5%). On each target, however, every one of its training seeds stays below every training seed of the target's own adversary. Adversaries nevertheless remain useful on planners other than their training target. In 19 of the 20 such cases, they cause more avoidable collisions than KING and ReGentS optimized for that planner, on each of their 3 training seeds. The exception is the Line-IDM adversary facing PDM, which exceeds ReGentS but not KING.

4.4 Additional experiments

To better understand Planner-Breaker, we train it with other collision criteria and without some of its components. We run these experiments with PDM and SP, which represent the rule-based and learned targets, with 3 training seeds per variant.

Table 3: **Variants of Planner-Breaker.** Target collisions with the PDM and SP targets for each variant. Bars as in Table 1. At-fault bars show the episodes with a target collision at fault under the nuPlan rule, ■ avoidable or ■ unavoidable. Means over 3 training × 3 evaluation seeds, and over 3 evaluation seeds for KING and ReGentS. †Training seeds differ widely (Table 15).

Target	PDM		SP	
	Collision class	At fault	Collision class	At fault
Planner-Breaker				
No driving reward				
At-fault reward				
Any-collision reward				
Termination on collision			 †	
KING				
ReGentS				

More collisions are not more informative. Rewarded for any target collision, the adversaries collide more often than KING and ReGentS, in 91.8% of episodes with PDM and 93.4% with SP (Table 3). However, 85.0 and 85.9% of episodes end in an unavoidable collision. They learn the easiest strategy, which is to ram the target, often from behind. Foreseeable collisions become rarer than with KING and ReGentS, at 1.5 and 3.8% of episodes, against 6.4 to 14.4%. These adversaries also do not specialize. They collide in 89 to 95% of episodes with every target, and not most often with their training target (Table 4). Rewarding collisions thus yields more collisions, but not more informative ones.

Planner-Breaker can optimize the at-fault rule. The at-fault rule of nuPlan is widely used to evaluate planners (Caesar et al., 2021; Karnchanachari et al., 2024; Dauner et al., 2024; Charraut et al., 2025). Its implementation in our simulator is detailed in Section A.4. Replacing our criterion with this rule in the adversarial reward is enough for Planner-Breaker to optimize it. With PDM, the adversaries then produce an avoidable collision in which the target is at fault in 64.9% of episodes (Table 3). Every other traffic we evaluated produces such collisions in at most 11.6% of episodes, and ordinary traffic in at most 0.9%. These adversaries also specialize, as each has its highest at-fault rate with its training target, on each training seed (Table 4).

The at-fault rule and avoidability disagree. The at-fault rule approximates avoidability from who hit whom. A target hit from behind is not at fault, which discards ramming. The rule does not discard the symmetric case, reverse ramming, since a moving target is at fault when an adversary reverses into its front. Adversaries trained against SP learn to exploit it. With SP, the target is at fault in 77.6% of episodes, but could have avoided the collision in only 33.8%. Conversely, the rule misses many collisions that the target could have foreseen and avoided. A target hit on its side while in its lane, as by a vehicle cutting in, is not at fault. With PDM, only 1.5% of the foreseeable

Table 4: **Specialization under other rewards.** Target collisions with each target (column) under the adversaries trained with the any-collision or at-fault reward against the planner of the row. Bars as in Table 3. Means over 3 training × 3 evaluation seeds.

Adv. \ Target	Line-IDM	IDM	PDM	SP	SP-Cond
Any-collision reward					
PDM					
SP					
At-fault reward					
PDM					
SP					

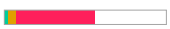
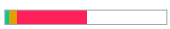
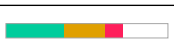
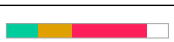
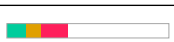
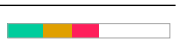
collisions caused by our adversaries are at fault. The adversary trained on SP under this rule also does not tell the two learned targets apart as well as ours (77.6 against 76.0% at fault, Section 4.3). Since avoidability is what matters for an automated vehicle, we argue that it should be measured with a counterfactual test, and optimized directly when possible, which Planner-Breaker enables.

The driving reward keeps adversaries driving like traffic. The adversaries also optimize the driving reward. Per step, they collect 74 to 146% of the driving reward of SP traffic, which has the same driving reward but no adversarial reward. Averaged over the episode, they drive at a similar speed (4.0 to 5.7 against 5.0 m/s) and reach 81 to 116% of its goal rate (Table 17). These statistics describe how the adversaries drive overall, not how they attack. Without the driving reward, the adversaries cause more foreseeable collisions on each training seed (67.8 and 73.2% of episodes with PDM and SP, against 53.4 and 65.1%), but many of them stop driving, as 56 and 42% average less than 2 m/s, against 8 and 24%. We keep it so that adversaries that do not attack still behave as traffic, at the cost of part of the foreseeable collisions.

Without post-collision continuation, adversaries do not learn to attack reliably. When a collision terminates the hitter's episode, the adversaries do not learn to attack reliably within our training budget. With PDM, they stop attacking on each training seed, and the target collides in 1.4% of episodes, as under ordinary traffic (Table 1). With SP, the three training seeds differ. One stops attacking (1.6% collisions), and the two others collide in 38.7 and 24.6% of episodes, against 84.5% with continuation (Table 15). This is consistent with the conflicting incentives that termination creates (Section 3.2).

4.5 Transfer to real-world scenes

Table 5: **Target collisions on real-world scenes, zoomed ×2.5.** Each bar spans 40% of 10,000 WOMD validation scenes, coloured as in Table 1. Our adversaries are deployed without retraining, and their bars are means over 3 training seeds.

Traffic \ Target	Line-IDM	IDM	PDM	SP	SP-Cond
Log replay					
IDM					
SP					
Planner-Breaker					

To test Planner-Breaker on real-world scenes, we deploy our adversaries without retraining on 10,000 validation scenes of the Waymo Open Motion Dataset (Ettinger et al., 2021), in which the target replaces the logged ego vehicle (Section B.3). On such scenes, planners are usually evaluated against log replay or rule-based traffic such as IDM, which both rarely reveal their failures. Rule-based traffic tends to yield to the target (Ransiek et al., 2025), and IDM collides with it in at most

2.9% of scenes. Logged vehicles do not react to the target, so most of the collisions they cause are unavoidable (79 to 90%, except with Line-IDM, Table 5). Planner-Breaker produces foreseeable collisions in 4.9 to 14.4% of scenes, more than both traffics with every target (at most 3.4%). WOMD scenes differ from our training scenes. Their road layouts are more complex than those of the eight CARLA maps, and vehicles start from their logged positions rather than from our placement. Our rates therefore remain well below those on CARLA (45.0 to 76.5%).

5 Conclusion

We introduced Planner-Breaker, which trains all surrounding traffic with self-play RL to find failures of a given planner. Reinforcement learning only requires a scalar reward, and self-play at scale provides billions of interactions to learn from. Together, they let the adversaries optimize criteria that gradient-based methods cannot, such as whether the target could have avoided a collision. Optimized for our criterion, each adversary learns failures specific to the planner it is trained against, which that planner could have foreseen and avoided. Rewarded for any collision or under the at-fault rule, the adversaries find few such failures. Since avoidability is what matters for an automated vehicle, we believe it should be measured with counterfactual tests, and optimized directly when possible.

Limitations and future work. Our criterion makes two simplifications. First, the counterfactual only tests braking. A collision that the target could have avoided only by steering counts as unavoidable, so our rates of avoidable collisions are conservative. Second, the foreseeability test is geometric, and only approximates what a planner can perceive. Both tests can be replaced, since Planner-Breaker only needs the criterion to be computed (Section 4.4). We encourage practitioners to define criteria for the failures they care about, such as collisions avoidable by steering. Third, training queries the planner at every simulation step, so its cost grows with the planner's inference time. Since our adversaries already cause avoidable collisions with planners they were not trained against (Section 4.3), fine-tuning an existing adversary on a new planner could reduce this cost, but this is yet to be explored.

AI use statement

In this work, we used generative AI tools to provide feedback on the research methodology and experiments, and to implement methods. We have not used generative AI tools to generate synthetic datasets, help develop theoretical models or conceptual frameworks, or interpret results. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, assisting with translation, cleaning and reformatting datasets, and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to create or edit software code, and to edit the paper to improve its readability. We have reviewed all AI-assisted work. We manually checked the code and are responsible for it. We used LLMs to improve the text at the sentence level, checking each sentence before adding it to the paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

The primary goal of adversarial driving is to increase safety on the roads, by generating collisions in simulation so that they do not happen in real life. One could imagine, though, that an evil person puts adversarial driving on the roads, which could lead to disastrous consequences. But if an evil person has enough power to control other people's cars, they will not need a state-of-the-art adversarial algorithm to create havoc.

References

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. NuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles, June 2021.

Xuan Cai, Xuesong Bai, Zhiyong Cui, Peng Hang, Haiyang Yu, and Yilong Ren. Adversarial Stress Test for Autonomous Vehicle via Series Reinforcement Learning Tasks With Reward Shaping. *IEEE Transactions on Intelligent Vehicles*, 10(2):832–847, February 2025. ISSN 2379-8904. doi: 10.1109/TIV.2024.3418887.

Valentin Charraut, Waël Doulazmi, Thomas Tournaire, and Thibault Buhet. V-Max: A RL Framework for Autonomous Driving. *Reinforcement Learning Journal*, 6:2427–2451, 2025.

Daphne Cornelisse, Spencer Cheng, Pragnay Mandavilli, Julian Hunt, Kevin Joseph, Waël Doulazmi, Valentin Charraut, Aditya Gupta, Joseph Suarez, and Eugene Vinitsky. PufferDrive: A fast and friendly driving simulator for training and evaluating RL agents, 2025a.

Daphne Cornelisse, Aarav Pandya, Kevin Joseph, Joseph Suárez, and Eugene Vinitsky. Building reliable sim driving agents by scaling self-play, May 2025b.

Daphne Cornelisse, Julian Hunt, Zixu Zhang, Waël Doulazmi, Kevin Joseph, Jaime Fernández Fisac, and Eugene Vinitsky. Human-like autonomy emerges from self-play and a pinch of human data, June 2026.

Marco Cusumano-Towner, David Hafner, Alex Hertzberg, Brody Huval, Aleksei Petrenko, Eugene Vinitsky, Erik Wijmans, Taylor Killian, Stuart Bowers, Ozan Sener, Philipp Krähenbühl, and Vladlen Koltun. Robust Autonomy Emerges from Self-Play, February 2025.

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with Misconceptions about Learning-based Vehicle Motion Planning. In *Proceedings of The 7th Conference on Robot Learning*, pp. 1268–1281. PMLR, December 2023.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-Driven Non-Reactive Autonomous Vehicle Simulation and Benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, December 2024. doi: 10.52202/079017-0902.

Daniel Dauner, Valentin Charraut, Bastian Berle, Tianyu Li, Long Nguyen, Jiabao Wang, Changhui Jing, Maximilian Igl, Holger Caesar, Boris Ivanovic, Yiyi Liao, Andreas Geiger, and Kashyap Chitta. 123D: Unifying Multi-Modal Autonomous Driving Data at Scale, May 2026.

Wenhao Ding, Baiming Chen, Minjun Xu, and Ding Zhao. Learning to Collide: An Adaptive Safety-Critical Scenarios Generating Method. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2243–2250, October 2020. doi: 10.1109/IROS45743.2020.9340696.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16. PMLR, October 2017.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R. Qi, Yin Zhou, Zoey Yang, Aurélien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large Scale Interactive Motion Forecasting for Autonomous Driving: The Waymo Open Motion Dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9710–9719, 2021.

Shuo Feng, Haowei Sun, Xintao Yan, Haojie Zhu, Zhengxia Zou, Shengyin Shen, and Henry Liu. Dense reinforcement learning for safety validation of autonomous vehicles. *Nature*, 615:620–627, March 2023. doi: 10.1038/s41586-023-05732-2.

Niklas Hanselmann, Katrin Renz, Kashyap Chitta, Apratim Bhattacharyya, and Andreas Geiger. KING: Generating Safety-Critical Driving Scenarios for Robust Imitation via Kinematics Gradients. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, volume 13698, pp. 335–352. Springer Nature Switzerland, Cham, 2022. ISBN 978-3-031-19838-0 978-3-031-19839-7. doi: 10.1007/978-3-031-19839-7_20.

Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, pp. 709–715, San Jose, California, July 2004. AAAI Press. ISBN 978-0-262-51183-4.

Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, Yiluan Guo, and Holger Caesar. Towards learning-based planning: The nuPlan benchmark for real-world autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 629–636, May 2024. doi: 10.1109/ICRA57147.2024.10610077.

Saman Kazemkhani, Aarav Pandya, Daphne Cornelisse, Brennan Shacklett, and Eugene Vinitsky. GPUDrive: Data-driven, multi-agent driving simulation at 1 million FPS. In *The Thirteenth International Conference on Learning Representations*, October 2024.

Mark Koren, Saud Alsaif, Ritchie Lee, and Mykel J. Kochenderfer. Adaptive Stress Testing for Autonomous Vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–7, June 2018. doi: 10.1109/IVS.2018.8500400.

Ritchie Lee, Ole J. Mengshoel, Anshu Saksena, Ryan W. Gardner, Daniel Genin, Joshua Silbermann, Michael Owen, and Mykel J. Kochenderfer. Adaptive Stress Testing: Finding Likely Failure Events with Reinforcement Learning. *Journal of Artificial Intelligence Research*, 69:1165–1201, December 2020. ISSN 1076-9757. doi: 10.1613/jair.1.12190.

Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning Proceedings 1994*, pp. 157–163. Elsevier, 1994. ISBN 978-1-55860-335-6. doi: 10.1016/B978-1-55860-335-6.50027-1.

Yuxin Liu, Zhenghao Peng, Xuanhao Cui, and Bolei Zhou. Adv-BMT: Bidirectional Motion Transformer for Safety-Critical Traffic Scenario Generation. https://arxiv.org/abs/2506.09485v2, June 2025.

Tong Nie, Yihong Tang, Junlin He, Yuewen Mei, Jie Sun, Lijun Sun, Wei Ma, and Jian Sun. ADV-0: Closed-Loop Min-Max Adversarial Training for Long-Tail Robustness in Autonomous Driving, March 2026.

OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with Large Scale Deep Reinforcement Learning, December 2019.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust Adversarial Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 2817–2826. PMLR, July 2017.

Joshua Ransiek, Johannes Plaum, Jacob Langner, and Eric Sax. GOOSE: Goal-Conditioned Reinforcement Learning for Safety-Critical Scenario Generation. In *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2651–2658, September 2024. doi: 10.1109/ITSC58415.2024.10919498.

Joshua Ransiek, Philipp Reis, Tobias Schürmann, and Eric Sax. Adversarial and Reactive Traffic Entities for Behavior-Realistic Driving Simulation: A Review, May 2025.

Davis Rempe, Jonah Philion, Leonidas J. Guibas, Sanja Fidler, and Or Litany. Generating Useful Accident-Prone Driving Scenarios via a Learned Traffic Prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17305–17315, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017. URL `http://arxiv.org/abs/1707.06347`. arXiv:1707.06347 [cs].

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a Formal Model of Safe and Scalable Self-driving Cars, October 2018.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, December 2018. doi: 10.1126/science.aar6404.

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*, 10(9):9320–9327, September 2025. ISSN 2377-3766. doi: 10.1109/LRA.2025.3592090.

Joseph Suarez. PufferLib 2.0: Reinforcement Learning at 1M steps/s. *Reinforcement Learning Journal*, 6:1378–1388, 2025.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning, second edition: An Introduction*. Bradford Books, Cambridge, Massachusetts, 2nd edition edition, November 2018. ISBN 978-0-262-03924-6. URL http://incompleteideas.net/book/the-book-2nd.html.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805–1824, August 2000. ISSN 1063-651X, 1095-3787. doi: 10.1103/PhysRevE.62.1805. URL https://link.aps.org/doi/10.1103/PhysRevE.62.1805.

Eugene Vinitsky, Yuqing Du, Kanaad Parvate, Kathy Jang, Pieter Abbeel, and Alexandre Bayen. Robust Reinforcement Learning using Adversarial Populations. https://arxiv.org/abs/2008.01825v2, August 2020.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354, November 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-1724-z.

Akifumi Wachi. Failure-Scenario Maker for Rule-Based Agent using Multi-agent Adversarial Reinforcement Learning and its Application to Autonomous Driving. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 6006–6012, Macao, China, August 2019. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-4-1. doi: 10.24963/ijcai.2019/832.

Yuan Yin, Pegah Khayatan, Eloi Zablocki, Alexandre Boulch, and Matthieu Cord. ReGentS: Real-World Safety-Critical Driving Scenario Generation Made Stable. In *ECCV 2024 Workshop on Multimodal Perception and Comprehension of Corner Cases in Autonomous Driving*, September 2024.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. *Advances in Neural Information Processing Systems*, 35:24611–24624, December 2022. doi: 10.52202/068431-1787.

Linrui Zhang, Zhenghao Peng, Quanyi Li, and Bolei Zhou. CAT: Closed-loop Adversarial Training for Safe End-to-End Driving. In *Proceedings of The 7th Conference on Robot Learning*, pp. 2357–2372. PMLR, December 2023.

A METHOD DETAILS

A.1 OBSERVATION FUNCTION

The observation of each adversary follows GIGAFLOW and is summarized in Table 6. All features are expressed in the adversary's egocentric frame and normalized. At each step, a random subset of the lane and boundary segments in range is kept. To identify the target, each observed vehicle carries a flag that is 1 for the target and 0 otherwise.

Table 6: Observation of an adversary. Lane and boundary segments are in range up to 200 m ahead, 60 m behind and 50 m to the side, and the kept ones are sampled at random at each step.

Group	Slots, range	Features
Ego	1	speed, width, length, steering angle, longitudinal and lateral acceleration, distance and angle to the lane centre, speed limit, time stopped
Goals	3	relative position of the route goals, set to 0 once reached
Vehicles	16 nearest, 200 m	relative position, length, width, relative heading, speed, time stopped, target flag
Lanes	49 of 70	segment midpoint, half-length, width, direction, road-graph distance to the next goal, absolute and relative to the ego's lane
Road boundaries	30 of 50	segment midpoint, half-length, width, direction
Traffic controls	4 nearest, 200 m	stop-line endpoints, type, light state

A.2 DRIVING REWARD

The driving reward has the terms of the GIGAFLOW reward (Cusumano-Towner et al., 2025), with fixed weights instead of randomized ones, and an additional overspeed penalty (Table 7). A collision with another vehicle or leaving the road ends the agent's episode, except when an adversary hits the target (Section 3.2). In the self-play training of SP, running a red light also ends the episode. So that adversaries can run red lights, but rarely, running a red light does not end their episode, and its penalty is three times larger. In the ablation without driving reward (Section 4.4), the collision, off-road and red-light penalties are kept, and all other terms are set to 0.

Table 7: Driving reward of an adversary, in the notation of GIGAFLOW. v is the speed (m/s), a and $\dot{a}$ the acceleration and jerk, θ_f the heading relative to the lane, x_f the distance to the lane centre, g the current goal, v_{lim} the speed limit, and $\Delta t = 0.1$ s.

Reward	Value
$R_{\text{goal}} = \mathbf{1}\big(\lVert x - g\rVert < \delta_{\text{goal}} \wedge (\text{intermediate goal} \vee \lvert v\rvert < v_{\text{goal}})\big)$	$\delta_{\text{goal}} = 2$, $v_{\text{goal}} = 3$
$R_{\text{collision}} = -(\alpha_{\text{collision}} + 0.1\lvert v\rvert)\,\mathbf{1}_{\text{collision}}$	$\alpha_{\text{collision}} = 1.5$
$R_{\text{off-road}} = -\alpha_{\text{boundary}}\,\mathbf{1}_{\text{boundary}}$	$\alpha_{\text{boundary}} = 1.5$
$R_{\text{stop-line}} = -\alpha_{\text{stop-line}}\,\mathbf{1}_{\text{stop-line violation}}$	$\alpha_{\text{stop-line}} = 3$
$R_{\text{comfort}} = -\alpha_{\text{comfort}}\big(\mathbf{1}_{\lvert a_{\text{long}}\rvert>3} + \mathbf{1}_{\lvert a_{\text{lat}}\rvert>3} + \mathbf{1}_{\lvert \dot{a}_{\text{long}}\rvert>5 \vee \lvert \dot{a}_{\text{lat}}\rvert>5}\big)$	$\alpha_{\text{comfort}} = 0.05$
$R_{\text{l-align}} = \alpha_{\text{l-align}}\Delta t\big(\min(\cos\theta_f, 0) + \alpha_{\text{vel-align}}\min(v\cos\theta_f, 0) + 0.0025(1 - \lvert\theta_f\rvert/(\pi/2))\big)$	$\alpha_{\text{l-align}} = 0.025$, $\alpha_{\text{vel-align}} = 1$
$R_{\text{l-center}} = -\alpha_{\text{l-center}}\Delta t\big(\mathbf{1}_{\cos\theta_f > 0.5}\lvert x_f\rvert - 0.05/\exp(\lvert x_f\rvert - 0.5)\big)$	$\alpha_{\text{l-center}} = 0.001$
$R_{\text{velocity}} = \alpha_{\text{velocity}}\Delta t\max(\cos\theta_f, 0)\,\mathbf{1}_{v>2.5}$	$\alpha_{\text{velocity}} = 2.5 \times 10^{-3}$
$R_{\text{reverse}} = -\alpha_{\text{reverse}}\Delta t\,\mathbf{1}_{v<0}$	$\alpha_{\text{reverse}} = 0.005$
$R_{\text{timestep}} = -\alpha_{\text{timestep}}\Delta t\,\mathbf{1}_{\lvert v\rvert>0 \vee \lvert a\rvert>0}$	$\alpha_{\text{timestep}} = 2.5 \times 10^{-5}$
$R_{\text{overspeed}} = -\alpha_{\text{overspeed}}\,\mathbf{1}_{v > v_{\text{lim}}+2}$	$\alpha_{\text{overspeed}} = 0.05$

A.3 POST-COLLISION CONTINUATION AND TARGET-FOCUSED EPISODES

After a target collision, the target is removed from the scene and the adversaries keep driving for 10 s, after which the episode ends.

Each training scene contains 2 to 8 vehicles, drawn at random, one of which is the target. The target is placed on a random lane of the map. One adversary is placed on a lane within 10 m of the target. The scene is resampled until the route of this adversary over its next 35 m shares a lane with, or crosses, the route of the target over its next 35 m, for up to 10 attempts. The other adversaries are placed within 35 m of the target. As in GIGAFLOW, vehicle dimensions are randomized, with a length between 0.8 and 7 m and a width between 0.8 and 2.7 m, and each vehicle starts at a random speed between 0 and the speed limit of its lane.

Besides a target collision, the episode ends when the target leaves the road, when no adversary has been within 35 m of the target for 1 s, or after 2560 steps (256 s). The evaluation protocol is given in Appendix B.

A.4 COLLISION CLASSIFICATION

At-fault rule. We use the at-fault rule of nuPlan. In the words of Karnchanachari et al. (2024), the planner is penalized for "collisions with stopped agents, collisions with agents in front of ego and collisions with agents in adjacent lanes while making a lane change", but not "for rear-end collisions or other agents colliding with the ego when it is stopped". Our implementation applies the following checks in order, and the first one that holds decides.

1. The target is stopped (speed of at most 0.2 m/s): not at fault.
2. The other vehicle is stopped: at fault.
3. The centre of the other vehicle lies within 30° of the backward direction, seen from the rear of the target: not at fault.
4. The front bumper of the target intersects the other vehicle: at fault.
5. The target is on no lane, or its body extends more than 2.05 m from the lane centre: at fault.
6. Otherwise: not at fault.

Avoidability. The class is computed from the last 8 s of the recorded scene. The candidates τ are tested in increasing order from $\mathrm{d}t = 0.1$ s, and the search stops at the first success. The limit is $\tau_{\max} = \rho + v/b + 0.1$ s, where v is the target's speed at $t_c - \tau$, $b = 5$ m/s^2 is the braking deceleration, $\rho = 1$ s is the response time, and 0.1 s is a margin. In the replay, the target decelerates at b along its recorded path, extended in a straight line beyond its position at t_c. The replay lasts until the target has stopped, and at least until t_c. Braking fails in three cases. The target overlaps the adversary involved in the collision at any step of the replay. The target overlaps another adversary while at fault. After the replay, the adversary involved in the collision, moving in a straight line at its velocity at t_c, would hit the stopped target at any later time.

Foreseeability. To be robust to dangers that appear and disappear between consecutive steps, the recorded scene is checked at every step within 0.2 s of $t_c - \tau^\star - \rho$, and the collision is foreseeable if the situation is dangerous at any of these steps. At each checked step, the danger horizon is $\rho + v/b + 0.1$ s, with v the target's speed at that step. The two projections are evaluated every 0.1 s. In the first, both vehicles move at constant velocity. In the second, the target follows its route at constant speed, and the adversary moves at constant velocity. In both projections and at the checked step itself, the bounding box of the target is widened on each side by a lateral margin $m = \min\left(2,\ 0.2 + v_\perp^2/(2 \cdot 0.8)\right)$ m, where $v_\perp$ is the lateral speed of the adversary towards the target, in the target's frame. This margin simplifies the lateral safe distance of RSS (Shalev-Shwartz et al., 2018). It uses the relative lateral speed instead of the speed of each vehicle, has no response-time term, and is capped at 2 m.

Reward values. Foreseeable collisions are rewarded with $+10$, unforeseeable ones with $+2$, unavoidable ones with -0.1, and no collision with 0. Unforeseeable collisions are rewarded for two reasons. Since foreseeable collisions are rare under random exploration and unforeseeable ones

are easier to reach, rewarding unforeseeable collisions teaches the adversaries that hitting the target pays. Moreover, a target that reacts faster than ρ could avoid some of them, so they rank above unavoidable collisions. Without a penalty for unavoidable collisions, adversaries could remove the target with an easy unavoidable collision and then collect driving reward during the continuation. The penalty stays small because early random collisions are mostly unavoidable, and a large penalty would teach the adversaries to avoid the target.

B EXPERIMENTAL DETAILS

B.1 ARCHITECTURE, ACTIONS AND TRAINING

The policy follows the architecture of GIGAFLOW. Each observation group of Table 6 is encoded by its own two-layer network of width 128, and the encodings of the elements of each set (vehicles, lanes, boundaries, traffic controls) are max-pooled. The pooled encodings are concatenated and passed through a 3-layer network of width 768 with GELU activations, shared by the actor and the critic, each of which then has one hidden layer of width 256. The policy has no recurrent state.

As in GIGAFLOW, the policy chooses one of 12 discrete actions, the combinations of a longitudinal jerk in $\{-15, -4, 0, 4\}$ m/s^3 and a lateral jerk in $\{-4, 0, 4\}$ m/s^3. The jerks are integrated into accelerations bounded to $[-5, 2.5]$ m/s^2 longitudinally and $[-4, 4]$ m/s^2 laterally, with a time step of 0.1 s. In evaluation, each vehicle executes the mean of the jerks weighted by the action probabilities, instead of a sampled action.

The adversaries are trained with PPO (Schulman et al., 2017), with the hyperparameters of Table 8. Training is distributed over GPUs, each of which simulates 24 environments of 1024 agents and collects 128 steps per agent between updates, that is 3,145,728 transitions per GPU per update.

Table 8: PPO hyperparameters, identical for all targets.

Hyperparameter	Value
Discount γ, GAE λ	0.999, 0.95
Optimizer	AdamW ($\beta = 0.9, 0.999$, weight decay 0.01)
Learning rate	5×10^{-4}, cosine annealing to 0
Clip, entropy and value coefficients	0.2, 0.01, 0.5
Max gradient norm	0.5
Minibatch size, update epochs	65,536, 3
Rollout length per update	128 steps
Advantage filtering (GIGAFLOW)	threshold at 1% of the moving-average maximum $\lvert A \rvert$
Precision	bfloat16
Training budget	20×10^9 steps

B.2 TARGET PLANNERS

Line-IDM and IDM. The rule-based targets drive along their route and treat red lights as stopped vehicles. They set their speed directly, instead of through the jerk dynamics, and decelerate at up to 5 m/s^2. Line-IDM and IDM use the Intelligent Driver Model (Treiber et al., 2000) with the parameters of Table 9, and differ in how they choose the leading vehicle. Line-IDM takes the nearest vehicle that overlaps a straight corridor ahead of it, as wide as the target and $\mathrm{clip}(5\,\mathrm{s} \cdot v, 20\,\mathrm{m}, 40\,\mathrm{m})$ long. IDM follows the IDM agents of nuPlan (Karnchanachari et al., 2024). It samples its route every 1 m over the same distance, and takes the first vehicle that overlaps a sample. As in nuPlan, the footprint of each other vehicle is extended in proportion to its speed, here along its own route by the distance it covers in 1.5 s.

PDM. PDM reimplements PDM-Closed (Dauner et al., 2023). It generates 15 candidates, from 3 lateral offsets of the route (0 and ± 1 m) and 5 target speeds (100, 75, 50, 25 and 1% of the speed limit). Each candidate follows an IDM speed profile and is rolled out every 0.1 s over a horizon $H = \max(4, v/5 + 0.5)$ s, while the other vehicles are forecast at constant velocity. Each candidate

receives the score

$$10 \min\Big(\frac{\mathrm{TTC}}{H}, 1\Big) + 3\,\frac{v'}{v_{\mathrm{lim}}} + \mathbf{1}_{\mathrm{centre}} - 48 \cdot \mathbf{1}_{\mathrm{TTC}<T_d},$$

where TTC is the smallest time to collision with a vehicle or a red light along the rollout, v' is the next speed of the candidate, set to 0 when a collision is forecast within 5 s, $\mathbf{1}_{\mathrm{centre}}$ is 1 for the zero offset, and $T_d = \max(2, v/5 + 0.5)$ s. PDM executes the best candidate that stays on the road over the horizon. If the time to collision of this candidate is below T_d, PDM brakes at 5 m/s^2 instead.

Table 9: Intelligent Driver Model parameters of the rule-based targets. PDM uses them for the speed profiles of its candidates. The desired speed is the speed limit of the lane, at most 20 m/s.

	Line-IDM	IDM	PDM
Maximum acceleration (m/s^2)	1	1	4
Comfortable deceleration (m/s^2)	5	5	5
Time headway (s)	1.5	1.5	2
Minimum gap (m)	1	1	2
Exponent δ	4	4	4

Learned targets. SP and SP-Cond are trained by self-play in PufferDrive for 100B steps on the same eight CARLA maps, with 1 to 120 agents per scene. Running a red light ends an agent's episode, and 2% of the agents brake at random moments, as the erratic drivers of GIGAFLOW. SP uses fixed reward weights, similarly to Cornelisse et al. (2025b). SP-Cond adds the reward conditioning and randomization of GIGAFLOW, and its erratic drivers that do not see other vehicles, for 2% of the agents.

B.3 EVALUATION PROTOCOL

Evaluation uses the eight training maps but places the adversaries farther from the target, and it removes the mechanisms that only serve training: the early end, the observation dropout and the erratic drivers (Table 10). As in training, the first adversary is placed on a route that meets the route of the target. Vehicles execute the probability-weighted mean action (Section B.1), and each run evaluates 10,000 episodes. On WOMD, the target replaces the logged ego vehicle of each scene, and the adversaries start from the logged positions of the other vehicles, without retraining. Since WOMD evaluation is deterministic, we use a single evaluation seed.

Table 10: Training and evaluation protocols. The time step is 0.1 s in all settings.

	Training	CARLA evaluation	WOMD evaluation
Scenes	8 CARLA maps	same 8 maps	10,000 validation scenes
Vehicles per scene	2–8, random	2–8, random	logged, up to 128
Adversary placement	first within 10 m, others within 35 m	first within 25 m, others within 50 m	logged positions
Vehicle size (m)	0.8–7 × 0.8–2.7	2–5.5 × 1.5–2.5	logged
Goal spacing	25–75 m	30–80 m	30 m
Maximum length	2560 steps (256 s)	2560 steps (256 s)	91 steps (9.1 s)
After a target collision	target removed, 10 s continuation	both vehicles stop, 10 s continuation	episode ends
Early end	yes	no	no
Observation dropout, erratic drivers	yes	no	no
Action	sampled	mean	mean
Episodes per run	–	10,000	10,000
Seeds	training 29, 4, 99	evaluation 29, 4, 99	evaluation 43

B.4 KING AND REGENTS

We reimplement KING and ReGentS from the public code of ReGentS, which includes a reproduction of KING, with a differentiable kinematic bicycle model in PyTorch. Both methods perturb the trajectories of adversaries in a reference scene. To build the reference scenes, we place the vehicles as in our CARLA evaluation and roll out all of them with the SP policy for 100 steps. These rollouts play the role of the logged WOMD scenes of ReGentS and of the regular traffic of KING.

Each method selects the adversaries to optimize with its own rule. KING applies its cost to the closest adversary. ReGentS excludes static adversaries, and adversaries that stay in a 45° sector behind the target for most time steps (Yin et al., 2024). When no candidate remains, the scene is not optimized and counts as an episode without collision. Vehicles that are not optimized are driven by IDM. At each optimization iteration, the target is simulated in closed loop, and gradients do not flow through the target planner. As in ReGentS, both methods use Adam with a learning rate of 10^{-3} and at most 500 iterations. The optimized actions are then executed in PufferDrive, which classifies the target collisions as for our adversaries.

KING optimizes 80 steps at 4 frames per second (20 s), and ReGentS the length of WOMD scenarios. Since both optimize whole trajectories, and their results degraded beyond 100 steps in our tuning, we optimize them over 100 steps (10 s). Our adversaries have no time budget, and Table 11 gives the time at which they cause the target collision.

Table 11: Time of the target collision caused by Planner-Breaker (s), over the episodes with a target collision, pooled over 3 training $\times$ 3 evaluation seeds.

Target	First quartile	Median	Third quartile	95th percentile
Line-IDM	6.9	11.6	16.1	106.2
IDM	6.3	10.1	18.0	101.6
PDM	6.1	9.2	18.3	112.1
SP	6.8	10.9	16.9	58.5
SP-Cond	6.3	10.0	17.0	68.6
All targets	6.4	10.5	17.1	89.0

C ADDITIONAL RESULTS

Table 12, Table 13, Table 14 and Table 16 give the numerical values of the bar charts of Table 1, Table 2, Table 3 and Table 4, with standard deviations. Table 15 gives the training seeds of the variant marked with † in Table 3.

Table 12: **Numerical values of Table 1.** Rates in % of 10,000 episodes on the CARLA evaluation benchmark, mean ± standard deviation. For fixed traffic, KING and ReGentS, the standard deviation is over 3 evaluation seeds. For Planner-Breaker, evaluation seeds are averaged within each training seed and the standard deviation is over 3 training seeds.

Traffic \ Target	Line-IDM	IDM	PDM	SP	SP-Cond
(a) Target collision rate					
IDM	17.84±0.09	6.95±0.02	4.32±0.08	6.65±0.15	6.78±0.33
SP	19.56±0.25	7.81±0.21	1.65±0.12	1.21±0.12	1.22±0.13
Random	46.43±0.21	43.00±0.34	22.99±0.25	28.81±0.70	28.35±0.79
KING	87.63±0.15	91.52±0.22	83.31±0.24	89.61±0.29	89.36±0.25
ReGentS	70.56±0.31	69.00±0.36	62.43±0.41	69.22±0.35	68.03±0.18
Planner-Breaker	92.04±0.62	85.95±0.59	80.72±1.90	84.45±3.16	85.27±0.32
(b) Avoidable collision rate (foreseeable + unforeseeable)					
IDM	14.24±0.22	0.88±0.01	0.18±0.05	3.75±0.23	4.60±0.37
SP	16.91±0.27	2.02±0.12	0.50±0.10	0.70±0.08	0.69±0.10
Random	17.98±0.36	10.57±0.36	1.58±0.07	8.36±0.15	9.15±0.66
KING	31.53±0.59	16.84±0.63	24.09±0.26	26.42±0.46	26.54±0.35
ReGentS	24.59±0.67	15.87±0.31	15.71±0.52	19.83±0.52	19.43±0.24
Planner-Breaker	86.60±0.83	64.36±5.33	70.24±3.83	75.68±4.19	76.75±1.24
(c) Foreseeable collision rate					
IDM	9.69±0.05	0.59±0.05	0.02±0.02	0.78±0.06	1.11±0.08
SP	12.83±0.25	1.23±0.04	0.07±0.04	0.33±0.08	0.36±0.06
Random	10.79±0.36	5.32±0.18	0.45±0.08	3.72±0.19	4.07±0.11
KING	17.27±0.25	9.32±0.53	9.88±0.34	14.38±0.15	14.50±0.36
ReGentS	15.51±0.56	10.13±0.33	6.38±0.22	13.08±0.12	12.53±0.20
Planner-Breaker	76.49±0.60	45.01±2.84	53.43±4.51	65.10±3.33	66.80±0.99
(d) Unforeseeable collision rate					
IDM	4.55±0.26	0.29±0.05	0.16±0.04	2.97±0.18	3.49±0.35
SP	4.08±0.05	0.78±0.09	0.44±0.08	0.37±0.04	0.33±0.05
Random	7.19±0.30	5.25±0.30	1.14±0.03	4.64±0.16	5.07±0.57
KING	14.27±0.54	7.51±0.10	14.20±0.12	12.03±0.33	12.04±0.02
ReGentS	9.08±0.13	5.74±0.12	9.33±0.30	6.75±0.52	6.90±0.29
Planner-Breaker	10.12±0.26	19.35±2.70	16.81±1.16	10.58±0.98	9.95±0.26
(e) Unavoidable collision rate					
IDM	3.60±0.13	6.07±0.01	4.14±0.11	2.90±0.36	2.17±0.10
SP	2.64±0.14	5.79±0.32	1.14±0.04	0.51±0.05	0.53±0.03
Random	28.45±0.25	32.43±0.28	21.41±0.19	20.45±0.65	19.20±0.14
KING	56.09±0.64	74.69±0.42	59.23±0.37	63.20±0.19	62.82±0.54
ReGentS	45.97±0.81	53.13±0.60	46.71±0.87	49.39±0.36	48.60±0.33
Planner-Breaker	5.44±0.24	21.60±5.09	10.48±1.93	8.77±1.24	8.52±0.93

Table 13: **Numerical values of Table 2.** Rates in % of 10,000 episodes on the CARLA evaluation benchmark, for the adversarial policy trained against the planner of the row, facing each target (column). KING and ReGentS are optimized for each target. For the adversaries, evaluation seeds are averaged within each training seed and the standard deviation is over 3 training seeds. For KING and ReGentS, it is over 3 evaluation seeds.

Adv. \ Target	Line-IDM	IDM	PDM	SP	SP-Cond
(a) Target collision rate					
Line-IDM	92.04±0.62	73.03±6.03	49.47±3.84	66.64±1.78	71.64±2.48
IDM	74.43±1.56	85.95±0.59	69.53±0.58	66.58±1.87	67.69±1.74
PDM	79.31±0.25	78.19±2.22	80.72±1.90	77.45±2.28	77.78±1.89
SP	76.86±3.69	81.63±2.90	68.42±6.25	84.45±3.16	84.47±2.64
SP-Cond	68.23±4.06	77.91±3.20	56.27±6.35	80.85±0.76	85.27±0.32
Mixed	78.05±0.37	76.68±0.24	64.58±1.42	77.67±0.48	78.79±0.44
KING	87.63±0.15	91.52±0.22	83.31±0.24	89.61±0.29	89.36±0.25
ReGentS	70.56±0.31	69.00±0.36	62.43±0.41	69.22±0.35	68.03±0.18
(b) Avoidable collision rate (foreseeable + unforeseeable)					
Line-IDM	86.60±0.83	27.64±5.33	21.81±1.95	42.43±4.07	49.34±5.22
IDM	56.95±3.54	64.36±5.33	45.95±5.00	41.89±3.30	44.83±3.01
PDM	67.66±1.52	34.43±5.05	70.24±3.83	60.53±3.71	61.37±3.38
SP	59.63±9.94	31.54±10.13	48.75±9.21	75.68±4.19	75.27±3.97
SP-Cond	42.02±0.67	22.26±3.26	30.64±5.27	70.42±1.67	76.75±1.24
Mixed	63.39±1.06	43.88±2.23	50.39±0.65	66.54±1.74	68.23±1.62
KING	31.53±0.59	16.84±0.63	24.09±0.26	26.42±0.46	26.54±0.35
ReGentS	24.59±0.67	15.87±0.31	15.71±0.52	19.83±0.52	19.43±0.24
(c) Foreseeable collision rate					
Line-IDM	76.49±0.60	19.63±2.89	9.99±0.67	36.83±4.06	42.87±5.17
IDM	35.50±2.49	45.01±2.84	24.65±2.28	30.64±3.22	30.41±2.47
PDM	41.95±2.29	22.12±2.43	53.43±4.51	39.84±2.60	40.57±1.71
SP	29.61±4.00	15.28±4.57	20.03±5.50	65.10±3.33	59.88±1.52
SP-Cond	21.74±0.89	10.71±2.51	12.08±5.33	60.57±1.97	66.80±0.99
Mixed	42.00±2.53	27.97±0.43	32.61±1.13	55.56±1.35	54.58±0.47
KING	17.27±0.25	9.32±0.53	9.88±0.34	14.38±0.15	14.50±0.36
ReGentS	15.51±0.56	10.13±0.33	6.38±0.22	13.08±0.12	12.53±0.20
(d) Unforeseeable collision rate					
Line-IDM	10.12±0.26	8.01±2.47	11.81±1.28	5.61±0.29	6.47±0.13
IDM	21.45±1.06	19.35±2.70	21.30±3.05	11.25±0.33	14.42±1.62
PDM	25.71±1.99	12.31±2.72	16.81±1.16	20.69±1.13	20.80±1.96
SP	30.02±6.71	16.26±5.75	28.72±3.77	10.58±0.98	15.39±2.51
SP-Cond	20.29±0.74	11.55±0.84	18.56±0.21	9.85±0.30	9.95±0.26
Mixed	21.39±1.46	15.91±1.92	17.78±0.89	10.97±0.40	13.65±1.15
KING	14.27±0.54	7.51±0.10	14.20±0.12	12.03±0.33	12.04±0.02
ReGentS	9.08±0.13	5.74±0.12	9.33±0.30	6.75±0.52	6.90±0.29
(e) Unavoidable collision rate					
Line-IDM	5.44±0.24	45.38±1.53	27.66±2.44	24.21±3.33	22.30±3.09
IDM	17.48±4.44	21.60±5.09	23.57±4.59	24.69±4.96	22.86±4.70
PDM	11.66±1.75	43.76±3.72	10.48±1.93	16.92±1.48	16.41±1.50
SP	17.23±6.31	50.09±8.56	19.67±3.02	8.77±1.24	9.20±1.47
SP-Cond	26.21±3.60	55.65±0.11	25.63±1.51	10.43±1.11	8.52±0.93
Mixed	14.66±0.70	32.80±2.09	14.20±1.81	11.14±2.07	10.56±2.03
KING	56.09±0.64	74.69±0.42	59.23±0.37	63.20±0.19	62.82±0.54
ReGentS	45.97±0.81	53.13±0.60	46.71±0.87	49.39±0.36	48.60±0.33

Table 14: **Numerical values of Table 3.** Rates in % of 10,000 episodes on the CARLA evaluation benchmark. At-fault and avoidable collisions are at fault under the nuPlan rule and not unavoidable. For the variants of Planner-Breaker, evaluation seeds are averaged within each training seed and the standard deviation is over 3 training seeds. For KING and ReGentS, it is over 3 evaluation seeds.

Variant	Collision	Foreseeable	Unforeseeable	Unavoidable	At fault	At fault ∧ avoidable
(a) PDM target						
Planner-Breaker	80.72±1.90	53.43±4.51	16.81±1.16	10.48±1.93	2.61±0.44	1.80±0.47
No driving reward	82.22±0.32	67.81±0.66	9.52±0.89	4.89±0.19	2.83±0.15	1.70±0.16
At-fault reward	83.39±1.43	10.43±0.23	60.53±2.70	12.44±1.34	66.82±2.32	64.91±2.85
Any-collision reward	91.82±0.84	1.47±0.37	5.37±1.39	84.99±1.21	2.86±0.44	0.95±0.06
Termination on collision	1.40±0.14	0.13±0.01	0.32±0.02	0.95±0.12	0.85±0.05	0.33±0.03
KING	83.31±0.24	9.88±0.34	14.20±0.12	59.23±0.37	7.40±0.15	6.67±0.19
ReGentS	62.43±0.41	6.38±0.22	9.33±0.30	46.71±0.87	5.30±0.18	4.34±0.25
(b) SPtarget						
Planner-Breaker	84.45±3.16	65.10±3.33	10.58±0.98	8.77±1.24	35.98±1.81	34.37±1.91
No driving reward	92.50±0.26	73.23±1.42	13.62±0.76	5.66±0.82	43.41±3.47	41.23±3.48
At-fault reward	89.04±0.59	23.80±1.86	13.49±1.20	51.75±2.49	77.60±0.77	33.82±2.99
Any-collision reward	93.44±0.64	3.82±0.28	3.73±0.70	85.89±0.36	24.94±0.35	3.88±0.23
Termination on collision	21.62±18.73	14.57±13.40	2.87±2.29	4.18±3.13	10.14±9.36	8.68±8.20
KING	89.61±0.29	14.38±0.15	12.03±0.33	63.20±0.19	31.49±0.09	12.96±0.34
ReGentS	69.22±0.35	13.08±0.12	6.75±0.52	49.39±0.36	36.87±0.49	11.98±0.56

Table 15: **Training seeds of the variant marked with † in Table 3.** Rates in % of episodes, averaged over 3 evaluation seeds of 10,000 episodes.

Target, variant, training seed	Collision	Foreseeable	Unforeseeable	Unavoidable	At fault	At fault ∧ avoidable
SP, Termination on collision, seed 29	38.70	27.34	4.56	6.80	19.64	17.06
SP, Termination on collision, seed 4	1.59	0.61	0.27	0.71	0.93	0.67
SP, Termination on collision, seed 99	24.57	15.76	3.79	5.02	9.86	8.30

Table 16: **Numerical values of Table 4.** Rates in % of 10,000 episodes on the CARLA evaluation benchmark, for the adversaries trained with the any-collision or at-fault reward against the planner of the row, facing each target (column). At-fault and avoidable collisions are at fault under the nuPlan rule and not unavoidable. Evaluation seeds are averaged within each training seed, and the standard deviation is over 3 training seeds.

Adv. \ Target	Line-IDM	IDM	PDM	SP	SP-Cond
(a) Target collision rate					
Any-collision, PDM	92.65±1.23	94.47±0.65	91.82±0.84	91.26±1.64	91.10±1.46
Any-collision, SP	89.58±0.57	95.28±0.84	92.27±1.29	93.44±0.64	93.21±0.63
At-fault, PDM	86.17±0.49	69.65±3.55	83.39±1.43	80.13±2.11	79.54±1.83
At-fault, SP	88.63±0.55	93.61±0.29	87.23±0.78	89.04±0.59	88.79±0.58
(b) Foreseeable collision rate					
Any-collision, PDM	7.97±0.84	2.65±0.40	1.47±0.37	5.45±0.81	6.57±0.89
Any-collision, SP	5.68±0.09	1.38±0.06	0.42±0.04	3.82±0.28	5.04±0.57
At-fault, PDM	13.72±0.55	5.25±0.35	10.43±0.23	19.31±1.69	18.64±1.68
At-fault, SP	11.30±1.01	5.22±1.42	2.96±1.07	23.80±1.86	22.64±1.40
(c) Unforeseeable collision rate					
Any-collision, PDM	8.70±1.63	3.32±0.31	5.37±1.39	5.33±1.09	6.32±1.43
Any-collision, SP	5.91±0.57	2.52±0.45	1.70±0.27	3.73±0.70	4.09±0.73
At-fault, PDM	60.03±1.50	15.88±2.51	60.53±2.70	42.40±1.26	41.99±1.44
At-fault, SP	16.12±3.11	15.05±1.27	13.75±2.24	13.49±1.20	18.18±1.07
(d) Unavoidable collision rate					
Any-collision, PDM	75.99±1.64	88.50±0.04	84.99±1.21	80.49±0.41	78.21±0.88
Any-collision, SP	77.99±0.60	91.38±0.44	90.15±0.98	85.89±0.36	84.08±0.83
At-fault, PDM	12.42±1.23	48.52±2.91	12.44±1.34	18.42±1.37	18.92±1.48
At-fault, SP	61.20±3.55	73.33±2.47	70.53±2.56	51.75±2.49	47.97±1.74
(e) At-fault collision rate					
Any-collision, PDM	11.16±1.19	5.40±1.98	2.86±0.44	20.09±3.74	18.94±3.64
Any-collision, SP	11.40±0.36	3.25±0.25	2.10±0.20	24.94±0.35	23.22±0.85
At-fault, PDM	53.63±0.58	15.12±0.52	66.82±2.32	35.26±2.73	33.24±1.95
At-fault, SP	20.61±2.33	18.30±2.03	13.62±2.40	77.60±0.77	75.98±0.36
(f) At-fault and avoidable collision rate					
Any-collision, PDM	8.42±0.54	2.55±0.21	0.95±0.06	4.17±0.41	4.85±0.64
Any-collision, SP	8.35±0.12	0.96±0.02	0.25±0.06	3.88±0.23	4.82±0.36
At-fault, PDM	51.41±0.44	12.73±1.19	64.91±2.85	32.90±3.11	30.91±2.45
At-fault, SP	17.94±2.35	15.87±2.06	11.59±2.47	33.82±2.99	36.76±2.08

C.1 DRIVING OF THE ADVERSARIES

Table 17 describes the driving of our adversaries on the CARLA evaluation benchmark. Collision, off-road and red-light rates are the shares of adversaries that collide with another adversary, leave the road or run a red light at least once in an episode, so they grow with the episode length, which we also report. Lane alignment is the share of time an adversary drives within $60°$ of the lane direction. Driving reward and goals are per 1k steps of an adversary, and moving is the share of adversaries whose average speed exceeds 2 m/s. For comparison, the table also reports SP traffic facing the same target, and the adversaries trained without driving reward (Section 4.4), whose logged driving reward is not comparable since part of its terms are scaled by the zero weight.

Table 17: **Driving of the adversaries**, each facing its training target, on the CARLA evaluation benchmark, next to SP traffic facing the same target. Rates are shares of adversaries per episode. For our adversaries, evaluation seeds are averaged within each training seed, and the standard deviation is over 3 training seeds. For SP traffic, it is over 3 evaluation seeds.

Target	Traffic	Driving reward per 1k steps	Goals per 1k steps	Speed (m/s)	Moving (%)	Lane-aligned (%)	Adv.–adv. collision (%)	Off-road (%)	Red light (%)	Episode length (s)
Line-IDM	Ours	4.75±0.74	7.36±0.15	4.04±0.13	69.53±7.75	84.61±0.20	0.42±0.02	1.87±0.06	5.94±2.63	44.45±1.84
	SP traffic	6.42±0.01	9.13±0.03	5.02±0.01	98.31±0.05	94.95±0.16	0.32±0.03	4.98±0.13	2.55±0.02	218.71±0.94
IDM	Ours	8.45±0.44	9.75±0.27	5.41±0.11	92.72±0.89	86.79±0.96	0.43±0.02	1.69±0.04	1.93±0.74	58.06±2.31
	SP traffic	6.49±0.02	9.13±0.03	5.02±0.01	98.23±0.08	94.52±0.18	0.36±0.04	5.31±0.12	2.83±0.16	228.91±0.79
PDM	Ours	9.56±0.45	10.60±0.38	5.66±0.16	92.38±1.54	87.67±0.30	0.49±0.14	1.84±0.31	1.49±0.13	69.85±5.92
	SP traffic	6.57±0.02	9.17±0.03	5.00±0.01	98.71±0.07	95.18±0.18	0.34±0.04	5.49±0.07	2.78±0.06	241.42±0.50
	No driving reward	–	0.30±0.06	2.29±0.08	44.25±2.42	80.72±1.99	0.55±0.11	1.81±0.06	2.02±0.25	68.20±0.82
SP	Ours	5.77±0.79	7.47±0.56	4.67±0.38	75.72±9.06	88.08±0.37	0.44±0.10	1.80±0.31	3.06±0.78	53.10±8.18
	SP traffic	6.52±0.02	9.15±0.03	5.06±0.01	98.58±0.09	95.60±0.20	0.34±0.03	5.01±0.14	2.47±0.12	234.88±0.80
	No driving reward	–	0.57±0.02	2.74±0.06	58.17±2.68	84.60±0.41	0.66±0.25	1.92±0.06	1.69±0.05	30.81±0.84
SP-Cond	Ours	6.12±0.39	7.88±0.53	4.73±0.23	77.87±3.11	87.64±0.05	0.45±0.04	2.23±0.36	3.31±0.52	52.19±1.20
	SP traffic	6.48±0.02	9.09±0.03	5.02±0.02	98.61±0.03	95.41±0.08	0.35±0.02	5.27±0.04	2.57±0.05	238.56±0.37

C.2 RELEASED ADVERSARY

We will release the weights of an adversary trained for 100B steps against the five targets, with one target sampled uniformly in each episode and otherwise the setup of Section B.1. Table 18 gives its results on the CARLA evaluation benchmark.

Table 18: **Released adversary** on the CARLA evaluation benchmark. Rates in % of 10,000 episodes with a target collision of each class, mean ± standard deviation over 3 evaluation seeds of one training seed.

Rate (%) \ Target	Line-IDM	IDM	PDM	SP	SP-Cond
Collision	86.18±0.06	83.99±0.51	76.25±0.28	85.64±0.43	86.27±0.36
Avoidable	77.90±0.01	68.06±0.50	68.06±0.46	80.36±0.31	81.49±0.13
Foreseeable	61.53±0.76	49.51±0.49	49.21±0.72	70.57±0.04	70.94±0.14
Unforeseeable	16.37±0.76	18.54±0.01	18.86±0.28	9.79±0.33	10.55±0.15
Unavoidable	8.28±0.07	15.93±0.42	8.18±0.20	5.28±0.13	4.77±0.31