

KING: Generating Safety-Critical Driving Scenarios for Robust Imitation via Kinematics Gradients

Niklas Hanselmann^{1,2,3}, Katrin Renz^{2,3}, Kashyap Chitta^{2,3},
Apratim Bhattacharyya^{2,3}, and Andreas Geiger^{2,3}

¹Mercedes-Benz AG R&D, Stuttgart ²University of Tübingen

³Max Planck Institute for Intelligent Systems, Tübingen

<https://lasnik.github.io/king/>

Abstract. Simulators offer the possibility of safe, low-cost development of self-driving systems. However, current driving simulators exhibit naïve behavior models for background traffic. Hand-tuned scenarios are typically added during simulation to induce safety-critical situations. An alternative approach is to adversarially perturb the background traffic trajectories. In this paper, we study this approach to safety-critical driving scenario generation using the CARLA simulator. We use a kinematic bicycle model as a proxy to the simulator’s true dynamics and observe that gradients through this proxy model are sufficient for optimizing the background traffic trajectories. Based on this finding, we propose KING, which generates safety-critical driving scenarios with a 20% higher success rate than black-box optimization. By solving the scenarios generated by KING using a privileged rule-based expert algorithm, we obtain training data for an imitation learning policy. After fine-tuning on this new data, we show that the policy becomes better at avoiding collisions. Importantly, our generated data leads to reduced collisions on both held-out scenarios generated via KING as well as traditional hand-crafted scenarios, demonstrating improved robustness.

1 Introduction

After years of steady progress, autonomous driving systems are getting closer to maturity [25]. Due to the high consequences of failure, they have to satisfy extraordinarily high standards of robustness in the face of unseen and safety-critical scenarios. However, real-world data collection and validation for these situations is dangerous and lacks the necessary scalability [29, 31]. These problems can be addressed with realistic simulation. Unfortunately, current simulators such as CARLA [16] are not only insufficient in terms of visual fidelity but also lack the necessary diversity of driving scenarios: there exists both an *appearance* and a *content gap* to the real world [26]. The content gap poses a major challenge in the adoption of driving agents trained in simulation using imitation learning (IL) or reinforcement learning (RL), which are often brittle to o.o.d. inputs under-represented during training [17]. In this work, we aim to address the content gap by improving the behavior of simulated background traffic agents.

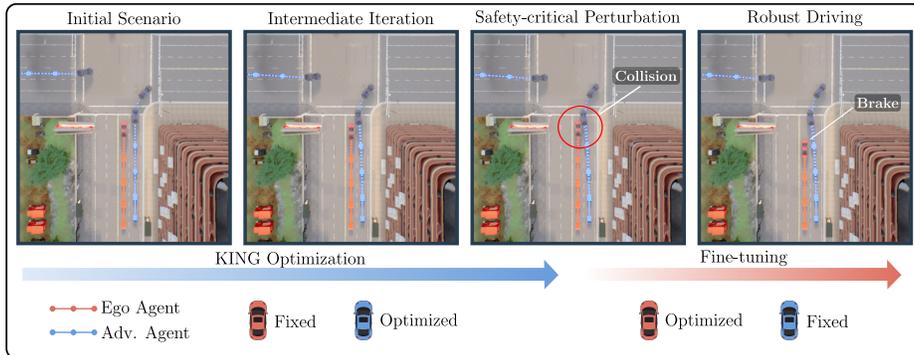


Fig. 1. Generating safety-critical scenarios for robust driving. Left: we propose KING, a novel optimization method to generate safety-critical driving scenarios which iteratively updates the initial scenario using gradients through a differentiable kinematics model and successfully induces a collision with the ego agent. Right: fine-tuning on expert behavior in safety-critical perturbations leads to a more robust agent.

Background agents in current simulators follow naïve behavioral models, resulting in limited diversity of the emerging traffic [16, 40]. Critical scenarios are often hand-crafted [1, 18]. This strategy is unlikely to be successful in fully covering the long-tailed distribution of critical situations that might be encountered in the real world. Furthermore, these scenarios are often non-adaptive to the driving agent under test. A more targeted approach is to actively seek possible failure modes. To do so, existing work perturbs the trajectories of background agents in a physically plausible manner to induce failures in the driving agent [2, 48]. This paradigm can be framed as a kinematically constrained adversarial attack on the driving agent, where the amount of safety-critical data generated within a given compute budget is dependent on the success rate of the attack. The prevalent approach for this task is black-box optimization (BBO), since simulators are often not differentiable [19, 23]. However, as we observe on the widely used CARLA simulator [16], existing attacks based on BBO (e.g. [19, 20, 23]) struggle to reliably induce collisions in IL-based driving agents (see Table 2).

As observed in image-space adversarial attacks, gradient-based optimization has the potential to be faster and more successful than BBO [3, 13]. Moreover, there has been a trend towards end-to-end differentiability, both in simulation [5, 44–46] and driving agents [6, 7, 10, 11, 36, 42]. Using differentiable components enables gradient-based generation of adversarial traffic scenarios. In this paper, we answer an important question: *does the entire simulation pipeline need to be differentiable to provide useful gradients for the optimization of traffic scenarios?* We present KING, a simple and effective approach for safety-critical scenario generation. Our key idea is to use a kinematic bicycle model as proxy to a driving simulator’s true dynamics, and solve for safety-critical perturbations of non-critical initial scenarios via backpropagation. The process of optimizing a non-critical scenario with KING is visualized in Fig. 1 (left). Further, we show

that KING generates challenging but solvable test cases for driving systems that use both (1) a planner that acts on a bird’s-eye view (BEV) grid input and (2) a camera and LiDAR-based driving agent [37]. Finally, we demonstrate that scenarios generated by KING can augment the original training distribution which has limited diversity. This leads to improved collision avoidance, as shown in Fig. 1 (right).

Contributions: (1) We propose KING, a simple procedure for generating safety-critical scenarios via backpropagation that is more reliable and requires less optimization time than BBO. (2) We show that KING generates challenging, diverse, and solvable scenarios for two different driving agents with different input modalities. (3) We use the generated scenarios to augment the CARLA simulator’s non-diverse traffic, improving the robustness of an end-to-end IL-based driving agent on both our generated test cases and a benchmark containing CARLA’s hand-crafted scenarios.

2 Related Work

End-to-End Driving: We are interested in stress-testing and improving end-to-end learning-based autonomous driving systems. While there are a few RL methods for this task [8, 47], most work leverages IL. Some adhere closely to the end-to-end learning paradigm [6, 11, 12, 30, 35, 37], directly inferring driving actions from raw sensor observations. However, others use interpretable intermediate representations [4, 43, 49]. In particular, BEV semantic occupancy grid representations are widely used in modern driving approaches [7, 10, 42, 51, 52]. This representation can be inferred from images [10, 21, 22, 27, 28, 32, 33, 41]. In our study we consider two IL-based driving agents reflecting both schools of thought: (1) a planner called AIM-BEV acting on ground-truth perception represented as a BEV semantic occupancy grid, and (2) an end-to-end agent acting on camera and LiDAR observations called TransFuser [37].

Generating Safety-Critical Scenarios: Previous work on generating safety-critical scenarios relies on BBO techniques and explores a variety of search space parameterizations, such as initial velocity or position of adversarial agents [14, 15, 38], a high-level route graph [2] or parameterized driving policies [31]. In AdvSim [48], the search space is parameterized as a sequence of kinematic bicycle model states for each adversarial agent, with steering and acceleration actions as free parameters. We also adopt this simple and expressive parameterization for KING. Different from this line of work, we propose a gradient-based procedure to optimize over these parameters rather than resorting to BBO techniques. Concurrent work presents STRIVE [39], a framework that also generates critical scenarios via gradient-based optimization. Here, an adversarial agent is parameterized as a latent vector of a learned motion forecasting model. STRIVE focuses on a simple, privileged rule-based planner rather than end-to-end IL agents and uses a proxy of the driving agent to enable gradient-based optimization, while KING directly optimizes for collisions wrt. the actual driving agent.

3 Safety-Critical Scenario Generation for Robust Imitation

In this section, we outline our overall approach for stress-testing and improving the robustness of IL-based driving agents, which is illustrated in Fig. 2. Given a driving agent trained on a dataset \mathcal{D}_{reg} of regular traffic, we propose KING, a novel gradient-based optimization procedure for automatically generating safety-critical perturbations of non-critical scenarios tailored to the agent under consideration. These scenarios serve to augment the original training distribution with limited diversity. In the following, we formally present our task settings, detail the parameterization and objective function used for scenario generation, and describe our robust training approach for IL.

Driving Agent and Regular Training: We assume that the driving policy of the agent is a neural network π_ω with parameters ω that takes in an observation $\mathbf{o}_t \in \mathbb{R}^{H_o \times W_o \times C_o}$ and goal location $\mathbf{x}_{goal} \in \mathbb{R}^2$ indicating the intended high-level route on the map, and plans a trajectory represented by four future 2D waypoints $\mathbf{w} \in \mathbb{R}^{4 \times 2}$:

$$\pi_\omega(\mathbf{o}_t, \mathbf{x}_{goal}) : \mathbb{R}^{H_o \times W_o \times C_o} \times \mathbb{R}^2 \rightarrow \mathbb{R}^{4 \times 2}. \quad (1)$$

Based on the predicted waypoints, the final actions $\mathbf{a}_t^0 \in [-1, 1]^2$ in the form of throttle and steering commands are produced by lateral and longitudinal controllers. Currently, several state-of-the-art IL agents fall under this paradigm [9, 10, 37]. With this general scheme, we consider both an IL policy with an intermediate representation as well as a strictly end-to-end model in our study. The first is a planner acting on ground-truth visual abstractions which we will refer to as AIM-BEV. This is inspired by [4] and the AIM-VA model in [10], but uses a BEV intermediate representation instead of 2D semantics since the BEV is an orthographic projection of the physical 3D space which is better correlated with vehicle kinematics than the 2D image domain. Here, the observations $\mathbf{o}_t \in \mathbb{R}^{192 \times 192 \times 3}$ are a rasterized BEV grid encoding HD map information with channels for (1) road and (2) lanes as well as a separate channel for dynamic obstacles such as background agents (3). The grid represents the environment ahead and to each side of the agent at a resolution of 5 pixels per meter. In addition to AIM-BEV, we also stress-test the publicly available checkpoint released by the authors of TransFuser [37]. This is a recent state-of-the-art IL-based self-driving model acting on observations $\mathbf{o}_t^{rgb} \in \mathbb{R}^{256 \times 256 \times 3}$ obtained from a front-facing camera and a discretized BEV lidar-histogram with two height bins $\mathbf{o}_t^{lid} \in \mathbb{R}^{256 \times 256 \times 2}$. Both AIM-BEV and TransFuser are trained on observation-waypoint pairs (\mathbf{o}, \mathbf{w}) drawn from \mathcal{D}_{reg} . The observations are mapped to a latent representation which is input to a gated recurrent unit (GRU) that plans the trajectory \mathbf{w} in an autoregressive fashion. For additional details, please refer to the supplementary material and the original TransFuser paper.

Gradient-based Scenario Generation: To optimize for safety-critical perturbations of an initial non-critical scenario (regular traffic), we iteratively simulate

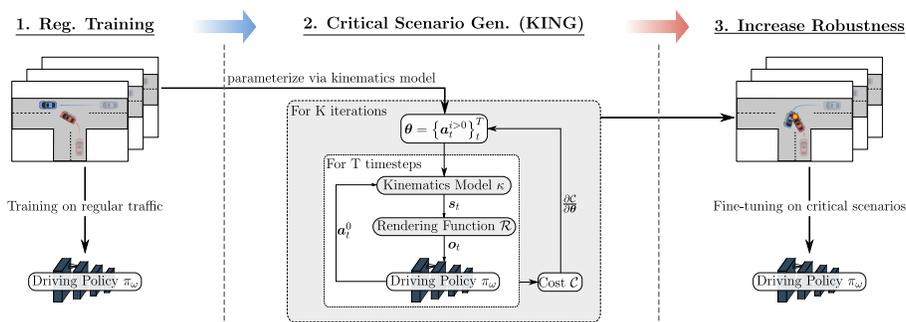


Fig. 2. Robust training pipeline. Given any agent with a driving policy π_ω trained on regular traffic data, we propose to increase its robustness under safety-critical scenarios by generating targeted augmentations. We propose KING, a gradient-based optimization procedure to obtain safety-critical perturbations of initial regular traffic scenarios. These perturbations then serve as additional training data for π_ω .

the scenario with the driving agent under attack (ego agent) in a closed-loop simulation. In particular, we aim to create a collision between the ego agent and one of the background actors (adversarial agents). At each iteration, we adjust the scenario’s parameters (i.e. the trajectories of adversarial agents) in order to induce such a collision. Importantly, the ego agent is able to react to the perturbations of the adversarial agents, since the attacks take place in a closed loop. Therefore, the scenarios generated are adaptive to the specific ego agent being attacked. In the following, we formally describe the simulation process and scenario generation procedure.

Let $\mathbf{x}_t^i \in \mathbb{R}^2$, $\psi_t^i \in [0, 2\pi]$ and $v_t^i \in \mathbb{R}$ be the ground-plane position, orientation and speed of the i -th agent at time t , where the index 0 indicates the ego agent. We denote the traffic state as $\mathbf{s}_t = \{\mathbf{x}_t^i, \psi_t^i, v_t^i\}_{i=0}^N$, where N is the number of agents. In slight abuse of notation, we will use \mathbf{s}_t^i to refer to the state of a specific agent. We instantiate a particular scenario as a sequence of these states $\mathcal{S} = \{\mathbf{s}_t\}_{t=0}^T$, where T is a fixed simulation horizon. \mathcal{S} is initialized using regular, non-critical traffic behavior as described in Section 4.2. To unroll the simulation forward in time, we compute the state at the next timestep \mathbf{s}_{t+1} given the current state \mathbf{s}_t and actions of all agents $\mathbf{a}_t = \{\mathbf{a}_t^i\}_{i=0}^N$ using the kinematics model κ , i.e., $\mathbf{s}_{t+1} = \kappa(\mathbf{s}_t, \mathbf{a}_t)$. We choose the bicycle model, which provides a strong prior on physically plausible motion of non-holonomic vehicles [8, 34] and is differentiable, enabling backpropagation through the unrolled state sequence \mathcal{S} . The ego agent is reactive to the simulation and chooses its actions \mathbf{a}_t^0 based on observations \mathbf{o}_t of the true underlying state, which are obtained through a rendering function \mathcal{R} , i.e., $\mathbf{o}_t = \mathcal{R}(\mathbf{s}_t, \mathcal{M})$. To render BEV semantic occupancy grids for AIM-BEV, we query a differentiable rasterizer [24] for the given current state \mathbf{s}_t and HD map \mathcal{M} , representing other agents by their bounding polygons. To render sensor data, e.g., camera and LiDAR data for TransFuser, we query the CARLA simulator’s

graphics engine. Note that all components of the simulation (π , κ and \mathcal{R}) are differentiable for AIM-BEV but \mathcal{R} is not differentiable for TransFuser.

Safety-Critical Perturbation: We perturb the sequence of states $\{\mathbf{s}_t^{i>0}\}_{t=0}^T$ for the N adversarial agents to induce a collision. If the ego agent collides within T timesteps, we terminate the simulation successfully. However, we would like the behavior of the adversarial agents to remain plausible. Hence, if any adversarial agent deviates from the drivable areas of the map or collides with another adversarial agent, the simulation terminates unsuccessfully. To detect collisions, we perform intersection checks between the bounding boxes of the agents. For out-of-bounds violations, we check if the adversarial agent bounding boxes enter the off-road area of the map.

As in [48], we parameterize the trajectories of adversarial agents as a sequence of states obtained by unrolling the kinematics model κ . Specifically, a safety-critical perturbation is found by optimizing the sequence of actions $\{\mathbf{a}_t^{i>0}\}_{t=1}^T$ for each adversarial agent. The overall search space can be written as $\boldsymbol{\theta} = \{\boldsymbol{\theta}^i\}_{i=1}^N$ where $\boldsymbol{\theta}^i = \{\mathbf{a}_{t=0}^i, \dots, \mathbf{a}_{t=T}^i\}$, with dimensionality $N \times T \times 2$. We optimize an objective \mathcal{C} which is motivated by prior work on safety-critical scenario generation [2, 15, 48]:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \mathcal{C}(\mathcal{S}) \quad \text{with} \quad \mathcal{C}(\mathcal{S}) = \phi_{col}^{ego}(\mathcal{S}) + \lambda \phi_{col}^{adv}(\mathcal{S}) + \gamma \phi_{dev}^{adv}(\mathcal{S}). \quad (2)$$

We encourage collisions involving the ego agent with the cost ϕ_{col}^{ego} , which measures the ℓ_2 distance to the closest adversarial agent. Unsuccessful terminations of the simulation are discouraged via the costs ϕ_{col}^{adv} and ϕ_{dev}^{adv} , weighted using hyper-parameters λ and γ . Here, ϕ_{col}^{adv} discourages collisions between adversarial agents by penalizing violations of a safety margin and ϕ_{dev}^{adv} penalizes deviations from the drivable area. These costs are similar to those commonly used in planning [7, 42, 50]. For details, we refer to the supplementary material.

Note that the realism of the generated scenarios is determined by the choice of regularizing terms in \mathcal{C} . While additional regularization may be beneficial, we find that the three terms in Eq. (2) are sufficient to find meaningful scenarios. We remark that our goal is to discover challenging scenarios that lie in the long tail of the distribution of traffic. Therefore, the scenarios discovered by our objective are not all likely to occur frequently in daily traffic. Importantly, however, the discovered scenarios are diverse, solvable, and enable learning more robust driving behaviors as demonstrated in Section 4.2 and Section 4.4.

Kinematics Gradients: Given that the sequence of states \mathcal{S} is unrolled based on the differentiable kinematics model, we can backpropagate costs at any timestep t to the set of actions $\{\mathbf{a}_{t-1}, \mathbf{a}_{t-2}, \dots, \mathbf{a}_0\}$ at previous timesteps. In the full unrolled computation graph of the simulation, the true gradients of the cost at any timestep can be taken wrt. the actions in preceding timesteps by recursively applying the chain rule along two paths: a direct path through the kinematics model and an indirect path, which additionally involves the driving policy π_ω and renderer \mathcal{R} . This is illustrated in Fig. 3.

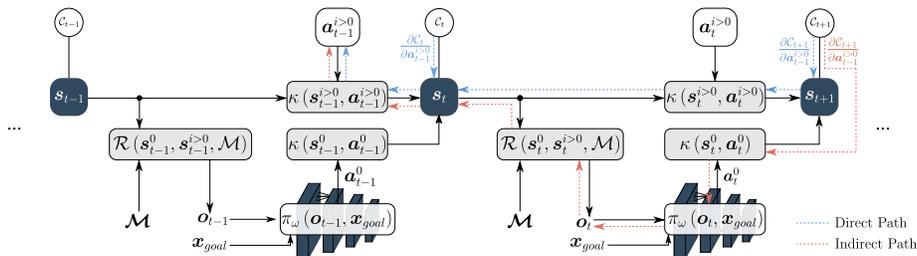


Fig. 3. Gradient paths. To unroll a simulation, we first render an observation \mathbf{o}_t of the traffic state \mathbf{s}_t using a rendering function \mathcal{R} . Both the ego agent policy π_ω and adversarial agents then take actions. The actions of the ego agent \mathbf{a}_t^0 depend on the observation and a goal location \mathbf{x}_{goal} . The actions of the adversarial agents $\mathbf{a}_t^{i>0}$ form the search space of the generation procedure. Given all actions, the next state \mathbf{s}_{t+1} is computed using a differentiable kinematics model κ . Gradients from the cost at time t can then be propagated back to preceding timesteps. The derivative has components along two paths: an efficient **direct path** and a compute-intensive **indirect path**.

With KING, we propose an approximation to the true gradients, which only considers the direct path and stops gradients through the indirect path. While this introduces an error in the gradient estimation, we empirically find it to work well while leading to several advantages. Firstly, as we show in Section 4.2, it enables gradient-based generation in the common case where the rendering function or driving policy is non-differentiable, preventing gradients to be taken wrt. the indirect path. Secondly, even when all components are differentiable, taking gradients wrt. to the indirect path involves backpropagating through the driving policy and rendering function (dotted red arrows in Fig. 3), incurring significant computational overhead. We investigate this setting for AIM-BEV where both the driving policy and rendering function are differentiable in Section 4.2 and show that given a fixed computational budget, this computational overhead leads to worse overall results compared to KING. We hypothesize that utilizing gradients through both paths becomes more important as the driving policy becomes robust to attacks.

Robust Training for IL: After stress-testing the IL-based driving agents, we are further interested in improving robustness by augmenting the original training data with the generated safety-critical scenarios. To this end, we pursue a simple yet effective strategy: (1) we generate a large set of safety-critical scenarios, (2) we filter these for scenarios in which a privileged rule-based expert algorithm finds a safe alternate trajectory, (3) we collect a dataset of observation-waypoint pairs \mathcal{D}_{crit} for the filtered scenarios using the expert, and (4) we fine-tune the policy π_ω with the standard L_1 loss \mathcal{L} on a mix of the safety-critical data \mathcal{D}_{crit} and the original dataset \mathcal{D}_{reg} :

$$\omega^* = \underset{\omega}{\operatorname{argmin}} \mathbb{E}_{(\mathbf{o}_t, \mathbf{x}_{goal}, \mathbf{w}) \sim (\mathcal{D}_{crit} \cup \mathcal{D}_{reg})} [\mathcal{L}(\mathbf{w}, \pi_\omega(\mathbf{o}_t, \mathbf{x}_{goal}))]. \quad (3)$$

4 Experiments

We begin by presenting the research questions we aim to answer in this study. **Can gradient-based attacks outperform black-box optimization (BBO) for safety-critical scenario generation?** We are interested in reducing the optimization time needed to take a set of non-critical scenario initializations and find interesting scenarios. Given the computational overhead of computing gradients and performing a backward pass, we analyze the gains that can be achieved for this task with gradient-based attacks over BBO in Section 4.2. In addition, as shown in Fig. 3, there are two paths for gradients through a simulator. We aim to understand the computational cost of backpropagating through each path and the corresponding gains in terms of collision rates.

Are gradient-based attacks applicable to non-differentiable simulators? While our main experiments are conducted using a differentiable simulator, in Section 4.3, we aim to investigate the applicability of KING to non-differentiable rendering functions, such as CARLA’s camera and LiDAR sensors.

Can we improve robustness by augmenting the training distribution with critical scenarios? We are interested in the analyzing robustness of the fine-tuned IL model that uses the data augmentation strategy described in Section 3. In Section 4.4, we investigate this on both the regular benchmark (hand-crafted scenarios) and held-out safety-critical test scenarios generated by KING.

4.1 Benchmarking IL Agents on Hand-Crafted Scenarios

To gain an initial understanding of their robustness, we first benchmark the agents used in our study with hand-crafted scenarios from CARLA. As an additional benchmark that aims to maximize the traffic interactions achievable with such scenarios, we select a set of short routes through intersections involving dense traffic. We describe these benchmarks below. The results provide a reference for performance of our AIM-BEV agent and the existing TransFuser agent on these settings which are relevant for the following experiments. All our experiments are conducted using CARLA version 0.9.10.1.

Experimental Setup: AIM-BEV and TransFuser [37] are trained via supervised learning to imitate a privileged expert on data containing regular CARLA traffic. The expert is a rule-based algorithm similar to the CARLA traffic manager autopilot. We evaluate these models on two benchmarks: (1) the NEAT validation routes from [10], and (2) a set of 82 routes through intersections in CARLA’s Town10 with dense traffic. The NEAT routes provide a holistic evaluation of the driving performance, but the evaluation is time-consuming. This set contains routes varying in length from 100m to 3km with regular CARLA traffic and hand-crafted scenarios. Since several of the routes are long and contain low traffic densities, poor collision avoidance has limited impact on the final metrics. For a more focused evaluation on collisions with traffic, the Town10 intersection routes are shorter in length (80m-100m). In this setting, we ensure a high density of dynamic agents by spawning vehicles at every possible spawn point permitted

Table 1. Performance on hand-crafted scenarios. We show the mean \pm std over 3 evaluations. AIM-BEV has fewer infractions than TransFuser on the NEAT validation routes. However, both agents collide in over 17% of the Town10 intersection routes.

Method	NEAT validation routes [10]				Town10 intersections			
	RC \uparrow	IS \uparrow	DS \uparrow	CR \downarrow	RC \uparrow	IS \uparrow	DS \uparrow	CR \downarrow
AIM-BEV	96.77 \pm 3.32	0.95 \pm 0.00	92.24 \pm 3.32	2.38 \pm 4.12	93.86 \pm 0.14	0.92 \pm 0.01	86.74 \pm 0.67	17.48 \pm 1.86
TransFuser [37]	99.25 \pm 1.30	0.78 \pm 0.03	77.59 \pm 2.01	11.90 \pm 4.12	93.68 \pm 2.01	0.85 \pm 0.00	80.03 \pm 0.79	17.48 \pm 0.70
Privileged Expert	99.83 \pm 0.07	1.00 \pm 0.00	99.83 \pm 0.07	0.00 \pm 0.00	94.89 \pm 0.33	0.97 \pm 0.00	92.81 \pm 0.53	3.66 \pm 0.00

by the CARLA simulator. Furthermore, each route is guaranteed to contain a hand-crafted scenario in which multiple vehicles enter the intersection from different directions at the same time. We selected Town10 for this benchmark as we found it to be the most challenging in preliminary experiments. On both of these benchmarks, we report the official metrics of the CARLA leaderboard, **Route Completion (RC)**, **Infraction Score (IS)** and **Driving Score (DS)**. RC is the percentage of the route completed by an agent before it gets blocked or deviates from the route. IS is a cumulative multiplicative penalty for every red light violation, stop sign violation, collision, and lane infraction. DS is the final metric, computed as the RC multiplied by the IS for each route. Each model is tested with three different evaluation seeds. In addition, we report the **collision rate (CR)**, which is the percentage of routes in which the agent was involved in a collision. Additional details regarding the driving metrics, rule-based expert, and training dataset for the driving policy are provided in the supplementary.

Results: The performance of both IL-agents as well as the rule-based expert which uses privileged information is shown in Table 1. Note that these methods have different inputs, and are not directly comparable. AIM-BEV achieves superior results in comparison to TransFuser. In particular, its significantly higher IS on the NEAT routes indicates that it is proficient at avoiding collisions when placed in sparse and non-adversarial CARLA traffic. On the Town10 intersections, AIM-BEV has a better IS than TransFuser, but we observe that the CR of both agents is similar (17.48%). This is much higher than the expert (CR=3.66%), showing that hand-crafted scenarios in dense traffic remain challenging for current IL-based methods. These hand-crafted scenarios are not adaptive to the agent, i.e., the same scenarios are applied for both AIM-BEV and TransFuser. In the following, we study the more targeted approach of actively generating safety-critical scenarios that are adaptive to the agent being attacked.

4.2 Comparison to BBO for Safety-Critical Scenario Generation

Next, we analyze the efficacy of KING for the generation of safety-critical scenarios, by comparing it with several BBO baselines for attacking AIM-BEV.

Experimental Setup: One scenario in our experimental setup involves rolling out a policy for 20 seconds of simulation time (80 timesteps at 4fps). We find this time horizon to be sufficient for the ego agent to traverse a route from

the start location to the end location while coming in close proximity to the adversarial agents. We compare several adversarial optimization techniques on 80 such scenarios. We obtain 4 maps (Town03-Town06) from the CARLA simulator. The 4 maps have a wide variety of road layouts, including intersections, single-lane roads, multi-lane highways, exits, and roundabouts. We sample a dense set of candidate start locations and end locations for the ego agent from the set of all junctions available in these 4 maps. The 80 ego agent routes in our evaluation are obtained by uniformly sampling 20 candidate routes per CARLA town. For each of these routes, we then initialize the adversarial agents to mimic regular CARLA traffic to obtain an initial, non-critical scenario, which allows explicit control over the traffic density. We use three traffic densities in our evaluation: 1, 2 and 4 agents (additional details in supplementary). The adversarial scenarios are evaluated using the **collision rate (CR)**, which is the percentage of routes for which the adversarial scenario search yielded a collision while respecting behavioral constraints. A search is only considered successful if all adversarial agents stay on drivable parts of the map (i.e., the road) and do not collide with other adversarial agents. To evaluate convergence, we report the average **time to 50% collision rate ($t_{50\%}$)**. This measures the average computation cost (in GPU seconds) required to find a collision in 50% of the total scenarios available. Finally, we report the runtime of each technique as the average number of optimization **seconds per iteration (s/it)**. The $t_{50\%}$ and s/it metrics for KING as well as all baselines are evaluated on a single RTX 2080Ti GPU. For all methods, we use a compute budget of 180 seconds per route on a single GPU, leading to a total experimental budget of up to 4 GPU hours for 80 routes.

Results: We now assess the efficiency of KING compared to BBO. To this end, we report the CR, $t_{50\%}$ and s/it of our approach and several baselines in Table 2. We consider the three traffic density settings separately, as well as the overall metrics for the complete set of 80×3 scenarios. Our baselines optimize the scenario parameters via BBO. In particular, besides **Random Search** and **Bayesian Optimization**, we consider **SimBA** [19], **CMA-ES** [20] and **Bandit-TD** [23]. SimBA is a variant of Random Search that greedily maximizes the objective and CMA-ES is a state-of-the-art evolutionary algorithm. Finally, Bandit-TD computes numerical gradients by integrating priors into a finite differences approach.

KING obtains a significantly higher CR than the BBO baselines in all 3 settings, increasing the number of scenarios for which a safety-critical perturbation is found by over 20%. Among the BBO baselines, CMA-ES attains the best overall scores with respect to both CR and $t_{50\%}$. Interestingly, the best performance for BBO is often observed for $N = 2$ agents. As we increase N from 1 to 2, it becomes easier for the baselines to find one nearby agent that can be perturbed to collide with the ego agent. However, further increasing N to 4 makes it harder to maintain plausible trajectories where the adversarial agents do not collide with each other or go off-road, leading to reduced performance. As the dimensionality of the search space increases (e.g. $N = 4$), KING begins to outperform the baselines in terms of $t_{50\%}$ by a large margin.

Table 2. Critical scenario generation on CARLA. We show the CR, $t_{50\%}$ and s/it for different optimization techniques in three traffic settings, as well as the aggregated metrics. KING finds collisions in over 80% of the initializations.

Method	1 Agent			2 Agents			4 Agents			Overall		
	CR \uparrow	$t_{50\%}$ \downarrow	s/it \downarrow	CR \uparrow	$t_{50\%}$ \downarrow	s/it \downarrow	CR \uparrow	$t_{50\%}$ \downarrow	s/it \downarrow	CR \uparrow	$t_{50\%}$ \downarrow	s/it \downarrow
Random Search	62.50	9.25	1.30	68.75	7.38	1.35	68.75	15.22	1.48	66.67	9.66	1.38
Bayesian Optimization	63.75	11.88	1.46	68.75	10.01	1.66	63.75	22.12	2.06	65.00	14.34	1.73
SimBA [19]	60.00	14.14	1.30	71.25	14.35	1.35	61.25	19.68	1.48	64.17	15.84	1.38
CMA-ES [20]	67.50	9.34	1.31	75.00	6.73	1.36	62.50	9.39	1.52	68.33	8.17	1.40
Bandit-TD [23]	37.50	-	3.87	30.00	-	4.39	21.25	-	5.02	29.58	-	4.43
KING Direct + Indirect	81.25	17.63	3.17	76.25	11.58	3.22	80.00	13.14	3.40	79.17	14.09	3.26
KING (Ours)	86.25	9.98	1.78	82.50	6.96	1.88	78.75	6.40	2.03	82.50	7.78	1.90

We also compare the proposed approximation in KING against the setting where we use gradients through entire simulation, including the driving policy and renderer (“KING Direct + Indirect” in Table 2). While also reliably finding safety-critical perturbations, the computational overhead of backpropagating through the indirect path leads to worse overall results given the same computation budget. This suggests the approximation in KING is reasonable for efficiently generating safety-critical scenarios. Additional results and details regarding the hyper-parameter choices for BBO are provided in the supplementary material. Since we observe that gradients through the direct path only are sufficient, we now conduct a detailed qualitative analysis where we apply KING to attack TransFuser, which requires the use of CARLA’s non-differentiable camera and LiDAR sensors for rendering.

4.3 Analysis of Safety-Critical Scenarios

In this section, we analyze the safety-critical scenarios generated by KING for both AIM-BEV and TransFuser in detail. Specifically, we show the distribution of the resulting scenarios with a traffic density of $N = 4$ agents in Fig. 4. For both driving agents, we first filter out the set of scenarios where KING is unable to find a collision (“No Collision”) as well as those that are not solvable by the rule-based expert (“Not Solvable”). We cluster the remaining scenarios using k-means (similar to [39]) to obtain 6 clusters of failure modes such as cut-ins (a_1), rear-ends (a_2) and unsafe behavior in unprotected turns (e,f). From the frequency of scenarios with “No collision” in Fig. 4, we observe that both AIM-BEV and TransFuser collide in at least 80% of the scenarios. This is a significant deviation from the collision avoidance of both models in the benchmarks shown in Table 1, where they attain a CR below 20%. The large amount of collisions for TransFuser indicates that KING can achieve promising results when applied out-of-the-box to driving simulators with non-differentiable rendering functions.

We show qualitative examples in Fig. 5 and 6, and additional examples in the supplementary material. Both AIM-BEV and TransFuser frequently collide in intersections when they encounter traffic that behaves differently from the traffic observed during training. Importantly, the “Not solvable” column shows

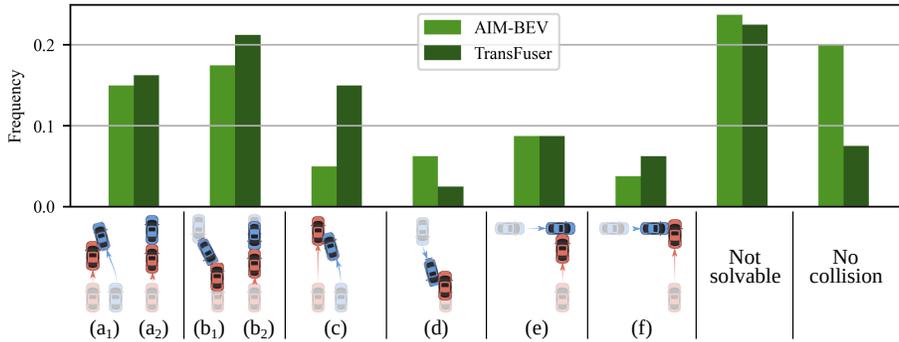


Fig. 4. Collision types. KING generates a diverse set of challenging but solvable scenarios. We group these into 6 clusters (a-f). The illustrations depict the ego agent in red and the adversarial agent in blue. The scenarios include (a) cut-ins and rear-ends caused by the ego agent, (b) head-on collisions, (c) merges, (d) side collisions with oncoming traffic, and t-bone collisions in intersections (e and f).

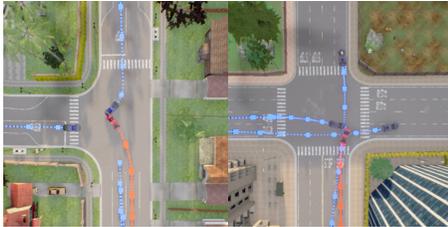


Fig. 5. KING scenarios - AIM-BEV. In intersections, AIM-BEV often fails to yield to the perturbed traffic. This leads to t-bone collisions, either by AIM-BEV (left) or the adversarial agent (right), corresponding to clusters (e) and (f) in Fig. 4.

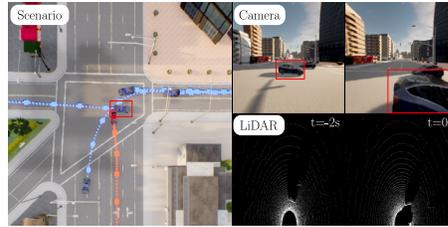


Fig. 6. KING scenario - TransFuser [37]. We show a scenario along with camera and LiDAR inputs two seconds before and at collision. TransFuser fails to slow down for an adversarial agent which stops inside the intersection (red box).

that for both agents, only around 20% of the scenarios have no feasible alternate trajectory. This leaves a large proportion of solvable scenarios in the 6 clusters shown in Fig. 4. The most frequent failure modes of both models are observed in clusters (a) and (b), which involve cut-ins, rear-ends, and head-on collisions. The rule-based expert solves these challenging scenarios by accurately forecasting the motion of the adversarial actors using privileged information. Interestingly, the failure cases are fairly evenly distributed over the 6 clusters which involve a wide variety of relative orientations between the colliding agents. The examples in Fig. 5 correspond to clusters (e) and (f). We highlight examples from clusters (a₁) and (a₂) for our experiment in Fig. 7. The high frequency and diversity of solvable scenarios generated by KING indicate its potential to augment the original training data for IL models, which we investigate next.

Table 3. Robust training for AIM-BEV. Results shown are the mean and std over 3 evaluation seeds. Fine-tuning with safety-critical scenarios reduces the CR by over 50% on other safety-critical scenarios as well as hand-crafted scenarios from CARLA.

Dataset	Held-out KING scenarios	Hand-crafted scenarios (Town10 intersections)			
	CR ↓	RC ↑	IS ↑	DS ↑	CR ↓
No Fine-tuning	100.00±0.00	93.86±0.14	0.92±0.01	86.74±0.67	17.48±1.86
\mathcal{D}_{reg}	57.14±0.00	95.66±0.51	0.90±0.00	86.85±0.62	19.51±0.00
\mathcal{D}_{crit}	28.57±0.00	91.92±0.19	0.96±0.00	88.37±0.41	6.10±0.00
$\mathcal{D}_{crit} \cup \mathcal{D}_{reg}$	28.57±0.00	94.42±0.36	0.96±0.36	90.20±0.00	8.13±0.70

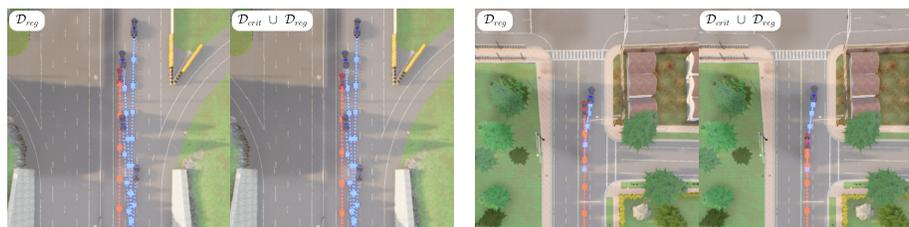


Fig. 7. Improved collision avoidance on held-out KING scenarios with AIM-BEV. Comparison of AIM-BEV before and after fine-tuning on $\mathcal{D}_{crit} \cup \mathcal{D}_{reg}$ in two KING scenarios. Ego agent in red, adversarial agent in blue. Best viewed zoomed in.

4.4 Evaluating Robustness after Fine-Tuning

Finally, we analyze the efficacy of the generated scenarios in augmenting the regular data \mathcal{D}_{reg} to yield more robust driving agents. Here, we evaluate robustness both wrt. safety-critical scenarios generated by KING and to hand-crafted scenarios in the CARLA simulator (using the Town10 intersections benchmark).

Experimental Setup: The goal of this experiment is to collect training data for improving collision avoidance. To this end, we build a large set of safety-critical scenarios by attacking AIM-BEV using initializations from Town03-Town06 of CARLA with $N = 4$ agents. To ensure meaningful supervision, we filter the resulting scenarios for ones where KING finds collisions that are solvable by the expert. This results in around 300 scenarios from which we hold out 20% for evaluation. We ensure that there is no overlap between the training and evaluation during this split by preventing routes with the same ego vehicle start location from being in both splits. Additional details regarding the training data and hyper-parameters are provided in the supplementary material.

Results: We report the driving performance of AIM-BEV after fine-tuning on $\mathcal{D}_{crit} \cup \mathcal{D}_{reg}$ in Table 3. Since the trajectories of the adversarial agents are fixed after optimization via KING, some of the scenarios may be solvable by simply adopting different overall driving styles, rather than becoming more proficient at collision avoidance. To quantify this, we fine-tune each model with only the original training data \mathcal{D}_{reg} as a baseline, which reduces the CR from 100% to 57.14% on the held-out KING scenarios. Additionally, we compare to fine-tuning

on only the critical scenarios \mathcal{D}_{crit} and the initial checkpoint from Table 1 (“No Fine-tuning”). Among the three fine-tuning strategies, using only \mathcal{D}_{reg} leads to unsatisfactory results, with a CR of 19.51% on the Town10 intersections benchmark. Using only \mathcal{D}_{crit} leads to a large reduction in CR on both evaluation settings. However, the model has a lower RC and only a small improvement in DS when compared to the \mathcal{D}_{reg} baseline on the Town10 intersections. Finally, using the combined dataset of $\mathcal{D}_{crit} \cup \mathcal{D}_{reg}$ gives the best results. In this setting, we obtain a CR of 28.57% on the KING scenarios, which is identical to the model fine-tuned with only \mathcal{D}_{crit} . However, the DS of this model on Town10 is improved by over 3 points, since it reduces the CR while maintaining a similar RC to the original model. This shows that the simple strategy of fine-tuning on a mixture of regular and safety-critical data is an effective way of learning from the scenarios generated by KING.

In Fig. 7, we show qualitative driving examples of the original and fine-tuned AIM-BEV agents on held-out KING scenarios, which belong to clusters (a₁) and (a₂) from Fig. 4. While these scenarios are straightforward to handle for an expert driver, AIM-BEV fails to brake for a vehicle stopping in between two lanes and is unable to maintain a safe distance in merging maneuvers, which highlights its brittleness in o.o.d scenarios. These scenario types do not frequently emerge naturally from the CARLA simulator’s background agent behavior which governs \mathcal{D}_{reg} . By incorporating data from \mathcal{D}_{crit} during training, the driving agent can learn to handle these scenarios safely.

5 Conclusion

In this work, we proposed a novel gradient-based generation procedure for safety-critical driving scenarios, KING, which achieves significantly higher success rates compared to existing BBO-based approaches while being more efficient. The key to our success is a compute-efficient direct gradient path through a kinematic motion model to guide the adversarial scenario generation process. Our analysis indicates that KING can achieve promising results when applied out-of-the-box to arbitrary driving agents. Furthermore, we show that despite having access to privileged BEV semantic maps as inputs, state-of-the-art IL-based driving policies are surprisingly brittle to minor perturbations in the behavior of the background actors. By augmenting their training data with scenarios from KING, we are able to significantly improve their collision avoidance. Exploring the robustness of agents with different training procedures (e.g. RL) offers an interesting direction for future research.

Acknowledgments: This work was supported by the BMWi (KI Delta Learning, project numbers: 19A19013A, 19A19013O), the BMBF (Tübingen AI Center, FKZ: 01IS18039A, 01IS18039B) and the DFG (SFB 1233, TP 17, project number: 276693517). We thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Katrin Renz and Kashyap Chitta. We also thank Aditya Prakash and Bernhard Jaeger for proofreading.

References

1. Carla autonomous driving leaderboard. <https://leaderboard.carla.org/> (2020) **2**
2. Abeyirigoonawardena, Y., Shkurti, F., Dudek, G.: Generating adversarial driving scenarios in high-fidelity simulators. In: Proc. IEEE International Conf. on Robotics and Automation (ICRA) (2019) **2, 3, 6**
3. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: A query-efficient black-box adversarial attack via random search. In: Proc. of the European Conf. on Computer Vision (ECCV) (2020) **2**
4. Behl, A., Chitta, K., Prakash, A., Ohn-Bar, E., Geiger, A.: Label efficient visual abstractions for autonomous driving. In: Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS) (2020) **3, 4**
5. Bergamini, L., Ye, Y., Scheel, O., Chen, L., Hu, C., Pero, L.D., Osinski, B., Grimmert, H., Ondruska, P.: Simnet: Learning reactive self-driving simulations from real-world observations. In: Proc. IEEE International Conf. on Robotics and Automation (ICRA) (2021) **2**
6. Bojarski, M., Testa, D.D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., Zieba, K.: End to end learning for self-driving cars. arXiv.org **1604.07316** (2016) **2, 3**
7. Casas, S., Sadat, A., Urtasun, R.: Mp3: A unified model to map, perceive, predict and plan. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2021) **2, 3, 6**
8. Chen, D., Koltun, V., Krähenbühl, P.: Learning to drive from a world on rails. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2021) **3, 5**
9. Chen, D., Zhou, B., Koltun, V., Krähenbühl, P.: Learning by cheating. In: Proc. Conf. on Robot Learning (CoRL) (2019) **4**
10. Chitta, K., Prakash, A., Geiger, A.: Neat: Neural attention fields for end-to-end autonomous driving. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2021) **2, 3, 4, 8, 9**
11. Codevilla, F., Miiller, M., López, A., Koltun, V., Dosovitskiy, A.: End-to-end driving via conditional imitation learning. In: Proc. IEEE International Conf. on Robotics and Automation (ICRA) (2018) **2, 3**
12. Codevilla, F., Santana, E., López, A.M., Gaidon, A.: Exploring the limitations of behavior cloning for autonomous driving. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2019) **3**
13. Deng, Y., Zheng, X., Zhang, T., Chen, C., Lou, G., Kim, M.: An analysis of adversarial attacks and defenses on autonomous driving models. In: Proc. of IEEE International Conference on Pervasive Computing and Communications (2020) **2**
14. Ding, W., Chen, B., Li, B., Eun, K.J., Zhao, D.: Multimodal safety-critical scenarios generation for decision-making algorithms evaluation. IEEE Robotics and Automation Letters (RA-L) **6**(2), 1551–1558 (2021) **3**
15. Ding, W., Xu, M., Zhao, D.: Learning to collide: An adaptive safety-critical scenarios generating method. In: Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS) (2020) **3, 6**
16. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Proc. Conf. on Robot Learning (CoRL) (2017) **1, 2**
17. Filos, A., Tigas, P., McAllister, R., Rhinehart, N., Levine, S., Gal, Y.: Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In: Proc. of the International Conf. on Machine learning (ICML) (2020) **2**

18. Fremont, D.J., Kim, E., Dreossi, T., Ghosh, S., Yue, X., Sangiovanni-Vincentelli, A.L., Seshia, S.A.: Scenic: A language for scenario specification and data generation (2022) [2](#)
19. Guo, C., Gardner, J.R., You, Y., Wilson, A.G., Weinberger, K.Q.: Simple black-box adversarial attacks. In: Proc. of the International Conf. on Machine learning (ICML) (2019) [2](#), [10](#), [11](#)
20. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation* **9**(2), 159–195 (2001) [2](#), [10](#), [11](#)
21. Hendy, N., Sloan, C., Tian, F., Duan, P., Charchut, N., Xie, Y., Wang, C., Philbin, J.: Fishing net: Future inference of semantic heatmaps in grids. arXiv.org **2006.09917** (2020) [3](#)
22. Hu, A., Murez, Z., Mohan, N., Dudas, S., Hawke, J., Badrinarayanan, V., Cipolla, R., Kendall, A.: FIERY: future instance prediction in bird’s-eye view from surround monocular cameras. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2021) [3](#)
23. Ilyas, A., Engstrom, L., Madry, A.: Prior convictions: Black-box adversarial attacks with bandits and priors. In: Proc. of the International Conf. on Learning Representations (ICLR) (2019) [2](#), [10](#), [11](#)
24. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: Advances in Neural Information Processing Systems (NIPS) (2015) [6](#)
25. Janai, J., Güney, F., Behl, A., Geiger, A.: Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art, vol. 12. Foundations and Trends in Computer Graphics and Vision (2020) [1](#)
26. Kar, A., Prakash, A., Liu, M., Cameracci, E., Yuan, J., Rusiniak, M., Acuna, D., Torralba, A., Fidler, S.: Meta-sim: Learning to generate synthetic datasets. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2019) [1](#)
27. Loukkal, A., Grandvalet, Y., Drummond, T., Li, Y.: Driving among Flatmobiles: Bird-Eye-View occupancy grids from a monocular camera for holistic trajectory planning. arXiv.org **2008.04047** (2020) [3](#)
28. Mani, K., Daga, S., Garg, S., Sai Shankar, N., Murthy Jatavallabhula, K., Madhava Krishna, K.: MonoLayout: Amodal scene layout from a single image. In: Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV) (2020) [3](#)
29. Norden, J., O’Kelly, M., Sinha, A.: Efficient black-box assessment of autonomous vehicle safety. arXiv.org **1912.03618** (2019) [1](#)
30. Ohn-Bar, E., Prakash, A., Behl, A., Chitta, K., Geiger, A.: Learning situational driving. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020) [3](#)
31. O’ Kelly, M., Sinha, A., Namkoong, H., Tedrake, R., Duchi, J.C.: Scalable end-to-end autonomous vehicle testing via rare-event simulation. In: Advances in Neural Information Processing Systems (NeurIPS) (2018) [1](#), [3](#)
32. Pan, B., Sun, J., Leung, H.Y.T., Andonian, A., Zhou, B.: Cross-view semantic segmentation for sensing surroundings. *IEEE Robotics and Automation Letters (RA-L)* (2020) [3](#)
33. Pillion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: Proc. of the European Conf. on Computer Vision (ECCV) (2020) [3](#)
34. Polack, P., Altché, F., d’Andréa Novel, B., de La Fortelle, A.: The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In: Proc. IEEE Intelligent Vehicles Symposium (IV) (2017) [5](#)

35. Pomerleau, D.: ALVINN: an autonomous land vehicle in a neural network. In: Advances in Neural Information Processing Systems (NIPS) (1988) [3](#)
36. Prakash, A., Behl, A., Ohn-Bar, E., Chitta, K., Geiger, A.: Exploring data aggregation in policy learning for vision-based urban autonomous driving. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020) [2](#)
37. Prakash, A., Chitta, K., Geiger, A.: Multi-modal fusion transformer for end-to-end autonomous driving. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2021) [3](#), [4](#), [8](#), [9](#), [12](#)
38. Priisalu, M., Pirinen, A., Paduraru, C., Sminchisescu, C.: Generating scenarios with diverse pedestrian behaviors for autonomous vehicle testing. In: Proc. Conf. on Robot Learning (CoRL) (2022) [3](#)
39. Rempe, D., Phillion, J., Guibas, L.J., Fidler, S., Litany, O.: Generating useful accident-prone driving scenarios via a learned traffic prior. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2022) [3](#), [11](#)
40. Richter, S.R., Hayder, Z., Koltun, V.: Playing for benchmarks. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2017) [2](#)
41. Roddick, T., Cipolla, R.: Predicting semantic map representations from images using pyramid occupancy networks. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020) [3](#)
42. Sadat, A., Casas, S., Ren, M., Wu, X., Dhawan, P., Urtasun, R.: Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In: Proc. of the European Conf. on Computer Vision (ECCV) (2020) [2](#), [3](#), [6](#)
43. Sauer, A., Savinov, N., Geiger, A.: Conditional affordance learning for driving in urban environments. In: Proc. Conf. on Robot Learning (CoRL) (2018) [3](#)
44. Scheel, O., Bergamini, L., Wolczyk, M., Osinski, B., Ondruska, P.: Urban driver: Learning to drive from real-world demonstrations using policy gradients. In: Proc. Conf. on Robot Learning (CoRL) (2021) [2](#)
45. Ścibior, A., Lioutas, V., Reda, D., Bateni, P., Wood, F.: Imagining the road ahead: Multi-agent trajectory prediction via differentiable simulation (2021) [2](#)
46. Suo, S., Regalado, S., Casas, S., Urtasun, R.: Trafficsim: Learning to simulate realistic multi-agent behaviors. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2021) [2](#)
47. Toromanoff, M., Wirbel, E., Moutarde, F.: End-to-end model-free reinforcement learning for urban driving using implicit affordances. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020) [3](#)
48. Wang, J., Pun, A., Tu, J., Manivasagam, S., Sadat, A., Casas, S., Ren, M., Urtasun, R.: Advsim: Generating safety-critical scenarios for self-driving vehicles. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2021) [2](#), [3](#), [6](#)
49. Xiao, Y., Codevilla, F., Pal, C., López, A.M.: Action-Based Representation Learning for Autonomous Driving. In: Proc. Conf. on Robot Learning (CoRL) (2020) [3](#)
50. Zeng, W., Luo, W., Suo, S., Sadat, A., Yang, B., Casas, S., Urtasun, R.: End-to-end interpretable neural motion planner. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019) [6](#)
51. Zhang, Z., Liniger, A., Dai, D., Yu, F., Van Gool, L.: End-to-end urban driving by imitating a reinforcement learning coach. In: Proc. of the IEEE International Conf. on Computer Vision (ICCV) (2021) [3](#)
52. Zhou, Y., Sun, P., Zhang, Y., Anguelov, D., Gao, J., Ouyang, T., Guo, J., Ngiam, J., Vasudevan, V.: End-to-end multi-view fusion for 3d object detection in lidar point clouds. In: Proc. Conf. on Robot Learning (CoRL) (2019) [3](#)