This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Vehicle trajectory prediction works, but not everywhere

Mohammadhossein Bahari*,1 Saeed Saadatnejad*,1 Ahmad Rahimi[2] Mohammad Shaverdikondori[2]
Amir Hossein Shahidzadeh[2] Seyed-Mohsen Moosavi-Dezfooli[3] Alexandre Alahi[1]

[1]EPFL Switzerland [2]Sharif university of technology [3] Imperial College London
{mohammadhossein.bahari, saeed.saadatnejad}@epfl.ch

Abstract

Vehicle trajectory prediction is nowadays a fundamental pillar of self-driving cars. Both the industry and research communities have acknowledged the need for such a pillar by providing public benchmarks. While state-of-the-art methods are impressive, i.e., they have no off-road prediction, their generalization to cities outside of the benchmark remains unexplored. In this work, we show that those methods do not generalize to new scenes. We present a method that automatically generates realistic scenes causing state-of-the-art models to go off-road. We frame the problem through the lens of adversarial scene generation. The method is a simple yet effective generative model based on atomic scene generation functions along with physical constraints. Our experiments show that more than 60% *of existing scenes from the current benchmarks can be modified in a way to make prediction methods fail (i.e., predicting off-road). We further show that the generated scenes (i) are realistic since they do exist in the real world, and (ii) can be used to make existing models more robust, yielding* $30-40\%$ *reductions in the off-road rate. The code is available online: https://s-attack.github.io/.*

1. Introduction

Vehicle trajectory prediction is one of the main building blocks of a self-driving car, which forecasts how the future might unfold based on the road structure (*i.e.,* the scene) and the traffic participants. State-of-the-art models are commonly trained and evaluated on datasets collected from a few cities [14, 19, 23]. While their evaluation has shown impressive performance, *i.e.*, almost no off-road prediction, their generalization to other types of possible scenes *e.g.*, other cities, remains unknown. Figure 1 shows a real-world example where a state-of-the-art model reaching zero off-road in the known benchmark [19] failed in *South St, New York, USA*. Since collecting and annotating data of all real-world scenes is not a viable and affordable solution, we present a method that automatically investigates the robustness of vehicle trajectory prediction to the scene. We tackle the problem through the lens of realistic adversarial scene generation.

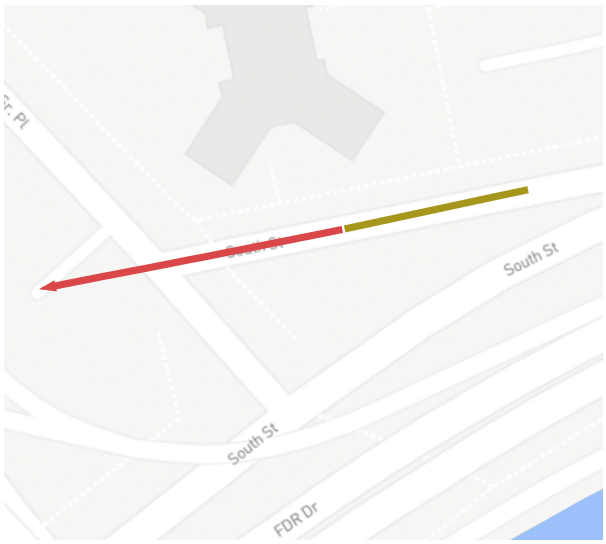


Figure 1. **A real-world place (location) in New York where the trajectory prediction model (here [32]) fails.** We find this place by retrieving real-world locations which resemble our conditional generated scenes for the prediction model.

Given an observed scene, we want to generate a realistic modification of it such that the prediction models fail in. Having an off-road prediction is a clear indication of a failure in the the model's scene reasoning and has been used in some previous works [8, 16, 36, 38]. To find a realistic example where the models go off-road, the huge space of possible scenes should be explored. One solution is data-driven generative models that mimic the distribution of a dataset [35]. Yet, they do not essentially produce realistic scenes due to the possible artifacts. Moreover, they will

* Equal contribution as the first authors.

represent a portion of real-world scenes as they cannot generate scenes beyond what they have observed in the dataset (cannot extrapolate). We therefore suggest a simple yet efficient alternative. We show that it is possible to use a limited number of simple functions for transforming the scene into new realistic but challenging ones. Our method can explicitly extrapolate to new scenes.

We introduce atomic scene generation functions where given a scene in the dataset, the functions generate multiple new ones. These functions are chosen such that they can cover a range of realistic scenes. We then choose the scenes where the prediction model produces an off-road trajectory. Using three state-of-the-art trajectory prediction models trained on Argoverse public dataset [19], we demonstrate that more than 60% of the existing scenes in the dataset can be modified in such a way that it will make state-of-the-art methods fail (*i.e.,* predict off-road). We confirm that the generated scenes are realistic by finding real-world locations that partially resemble the generated scenes. We also demonstrate off-road predictions of the models in those locations. To this end, we extract appropriate features from each scene and use image retrieval techniques to search public maps [1]. We finally show that these generated scenes can be used to improve the robustness of the models.

Our contributions are fourfold:

- we highlight the need for a more in-depth evaluation of the robustness of vehicle trajectory prediction models;
- our work proposes an open-source evaluation framework through the lens of realistic adversarial scene generation by promoting an effective generative model based on atomic scene generation functions;
- we demonstrate that our generated scenes are realistic by finding similar real-world locations where the models fail;
- we show that we can leverage our generated scenes to make the models more robust.

2. Related work

Vehicle trajectory prediction. The scene plays an important role in vehicle trajectory prediction as it constrains the future positions of the agents. Therefore, modeling the scene is common in spite of some human trajectory prediction models [13, 39]. In order to reason over the scene in the predictions, some suggested using a semantic segmented map to build circular distributions and outputting the most probable regions [21]. Another solution is reasoning over raw scene images using convolutional neural networks (CNN) [31]. Many follow-up works represented scenes in the segmented image format and used the learning capability of CNNs over images to account for the scene [10, 17, 18, 25, 40]. Carnet [45] used attention mechanism to determine the scene regions that were attended more, leading to an interpretable solution. Some recent work showed that scene can be represented by vector format instead of images [7, 24, 32, 47]. To further improve the reasoning of the model and generate predictions admissible with respect to the scene, use of symmetric cross-entropy loss [38, 41], off-road loss [8], and REINFORCE loss [16] have been proposed. Despite all these efforts, there has been limited attention to assess the performance of trajectory prediction models on new scenes. Our work proposes a framework for such assessments.

Evaluating self-driving systems. Self-driving cars deal with dynamic agents nearby and the static environment around. Several works studied the robustness of self-driving car modules with respect to the status of dynamic agents on the road, *e.g.*, other vehicles. Some previous works change the behavior of other agents in the road to act as attackers and evaluate the model's performance with regards to the interaction with other agents [3, 4, 20, 26, 28, 30, 43, 52]. Others directly modify the raw sensory inputs to change the status of the agents in an adversarial way [15, 49, 51, 53].

In addition to the dynamic agents, driving is highly dependant on the static scene around the vehicle. The scene understanding of the models can be assessed by modifying the input scene. Previous works modify the raw sensory input by changing weather conditions [33, 50, 54], generating adversarial drive-by billboards [29, 55], and adding carefully crafted patches/lines to the road [12, 46]. These works have not changed the shape of the scene, *i.e.*, the structure of the road. In contrast, we propose a conditional scene generation method to assess the scene reasoning capability of trajectory prediction models. Also our approach is different from data-driven scene generation based on graph [35] or semantic maps [44]. Data-driven generative models are prone to have artifacts and cannot extrapolate beyond the training data. Ours is an adversarial one which can extrapolate to new scenes.

3. Realistic scene generation

In this section, we explain in detail our approach for generating realistic scenes. After introducing the notations in Section 3.1, we show how we generate each scene in Section 3.2 and satisfy physical constraints in Section 3.3. Finally, we introduce our search method in Section 3.4.

3.1. Problem setup

The vehicle trajectory prediction task is usually defined as predicting the future trajectory of a vehicle z given its observation trajectory h, status of surrounding vehicles a, and scene S. For the sake of brevity, we assume S is in

the vector representation format [19][1] . Specifically, S is a matrix of stacked $2d$ coordinates of all lanes' points in x-y coordinate space where each row represents a point $s = (s_x, s_y)$. Formally, the output trajectory z of the predictor g is:

$$z = g(h, S, a). \tag{1}$$

Given a scene S, our goal is to create challenging realistic scene S^* as we will explain in Section 3.2.

3.2. Conditional scene generation

Our controllable scene generation method generates diverse scenes conditioned on existing scenes. Specifically, we opt for a set of atomic functions which represent turn as a typical road topology. To this end, we normalize the scene (*i.e.*, translation and rotation with respect to h), apply the transformation functions, and finally denormalize to return the generated scene to the original view. Note that every transformation of S is followed by the same transformations on h and a.

We define transformations on each scene point in the following form:

$$\tilde{s} = (s_x, s_y + f(s_x - b)) \tag{2}$$

where $\tilde{s}$ is the transformed point, f is a single-variable transformation function, and b is the border parameter that determines the region of applying the transformation. In other words, we define $f(< 0) = 0$ so the areas where $s_x < b$ are not modified. This confines the changes to the regions containing the prediction. One example is shown in Figure 2. The new scene is named $\tilde{S}$, a matrix of stacked $\tilde{s}$. We propose three interpretable analytical functions for the choice of f.

Smooth-turn: this function represents different types of single turns in the road.

$$\begin{aligned} f_{st,\alpha}(s_x) &= \begin{cases} 0, & s_x < 0 \\ q_\alpha(s_x), & 0 \le s_x \le \alpha_1 \\ (s_x - \alpha_1) q'_\alpha(\alpha_1) + q_\alpha(\alpha_1) & \alpha_1 < s_x \end{cases}, \\ q_\alpha(s_x) &= \alpha_2 s_x^{\alpha_3}, \\ \alpha &= (\alpha_1, \alpha_2, \alpha_3), \end{aligned} \tag{3}$$

where α_1 determines the length of the turn, α_2, α_3 control its sharpness, and q'_α indicates the derivative of the defined auxiliary function q_α. Note that according to the definition, $f_{st,\alpha}$ is continuously differentiable and makes a smooth turn. One such turn is depicted in Figure 2b.

[1] We show in the appendix that our method is seamlessly applicable when S is in image representation.

Double-turn: these functions represent two consecutive turns with opposite directions. Also, there is a variable indicating the distance between them:

$$\begin{aligned} f_{dt,\beta}(s_x) &= f_{st,\beta_1}(s_x) - f_{st,\beta_1}(s_x - \beta_2), \\ \beta &= (\beta_{11}, \beta_{12}, \beta_{13}, \beta_2), \\ \beta_1 &= (\beta_{11}, \beta_{12}, \beta_{13}), \end{aligned} \tag{4}$$

where β_1 is the set of parameters of each turn described in Equation (3) and β_2 is the distance between two turns. One example is shown in Figure 2c.

Ripple-road: one type of scene that can be challenging for the prediction model is ripple road:

$$\begin{aligned} f_{rr,\gamma}(s_x) &= \begin{cases} 0, & s_x < 0 \\ \gamma_1(1 - cos(2\pi\gamma_2\, s_x)), & s_x \ge 0 \end{cases}, \\ \gamma &= (\gamma_1, \gamma_2), \end{aligned} \tag{5}$$

where γ_1 determines the turn curvatures and γ_2 determines the sharpness of the turns. One such turn is depicted in Figure 2d.

3.3. Physical constraints

Every scenario consists of a scene and vehicle trajectories in it. The generated scenarios must be feasible, otherwise, they cannot represent possible real-world cases. We consider a scenario as feasible if a human driver can pass it safely. This means that the physical constraints – *i.e.*, the Newton's law – should not be violated. The Newton's law indicates a maximum feasible speed for each road based on its curvature [22]:

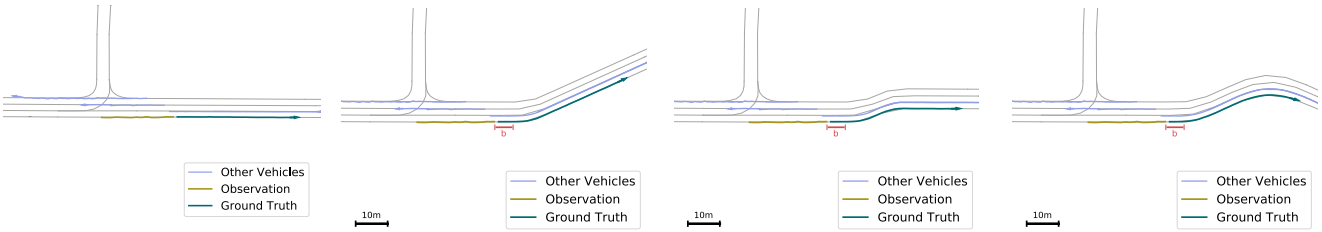
$$v_{max} = \sqrt{\mu g R}, \tag{6}$$

where R is the radius of the road, μ is the friction coefficient and g is the gravity. To consider the most conservative situation, we pick the maximum curvature (minimum radius) existing in the generated road. Then, we slow down the history trajectory when the speed is higher than the maximum feasible speed, and we name it $\tilde{h}$. Note that this conservative speed scaling ensures a feasible acceleration too. We will show in Section 4 that a model with hard-coded physical constraints successfully predicts the future trajectory for the generated scenes, which indicates that our constraints are enough.

3.4. Scene search method

In the previous sections, we defined a realistic controllable scene generation method. Now, we introduce a search method to find a challenging scene specific to each trajectory prediction model.

We define m as a function of z and S measuring the percentage of prediction points that are off-road obtained using a binary mask of the drivable area. We aim to solve the



(a) Before transformation (b) Single-turn (c) Double-turn (d) Ripple-road

Figure 2. **Visualization of different transformation functions**. The scene before transformation will be followed by three different transformations. Here, $\alpha = (10, 0.002, 3)$ for the single-turn, $\beta = (10, 0.002, 3, 10)$ for the double-turn and $\gamma = (6, 0.017)$ for the ripple-road. b is the border parameter and set to 5 meters in all figures.

following problem to find a scene in which the prediction model fails in:

$$S^* = \arg\min_{\tilde{S}} l(\tilde{z}, \tilde{S}),$$
$$l(\tilde{z}, \tilde{S}) = \Big(1 - m(\tilde{z}, \tilde{S})\Big)^2, \tag{7}$$

where $\tilde{S}$ is a modification of S according to Equation (2) using one of the transformation functions Equation (3), Equation (4), or Equation (5). Moreover, $\tilde{z} = g(\tilde{h}, \tilde{S}, \tilde{a})$ is the model's predicted trajectory given the modified scene and the modified history trajectories.The optimization problem finds the corresponding parameters to obtain S^* that gives the highest number of off-road prediction points. Equation (7) can be optimized using any black-box optimization technique. We have studied Bayesian optimization [42, 48], Genetic algorithms [5, 34], Tree-structured Parzen Estimator Approach (TPE) [9] and brute-force. The overall algorithm is described in the appendix.

4. Experiments

We conduct experiments to answer the following questions: 1) How is the performance of the prediction models on our generated scenes? 2) Are the generated scenes realistic and possibly similar to the real-world scenes? 3) Are we able to leverage the generated scenes to improve the robustness of the model?

4.1. Experimental setup

4.1.1 Baselines and datasets

We conduct our experiments on the baselines with different scene reasoning approaches (lane-graph attention [32], symmetric cross entropy [38], and counterfactual reasoning [27]), which are among the top-performing models and are open-source.

LaneGCN [32]. It constructs a lane graph from vectorized scene and uses self-attention to learn the predictions. This method was among the top methods in Argoverse Forecasting Challenge 2020 [2]. It is a multi-modal prediction model which also provides the probability of each mode. Therefore, in our experiments, we consider the mode with the highest probability.

DATF [38]. It is a flow-based method which uses a symmetric cross-entropy loss to encourage producing on-road predictions. This multi-modal prediction model does not provide the probability of each mode. We therefore consider the mode which is closest to the ground truth.

WIMP [27]. They employ a scene attention module and a dynamic interaction graph to capture geometric and social relationships. Since they do not provide probabilities for each mode of their multi-modal predictions, we consider the one which is closest to the ground truth.

MPC [6, 56]. We report the performance of a rule-based model with satisfied kinematic constraints. We used a well-known rule-based model which follows center of the lanes [56]. While many approaches can be used to satisfy kinematic constraints in trajectory prediction, similar to [6], we used Model Predictive Control (MPC) with a bicycle dynamic model.

We leveraged Argoverse dataset [19], the same dataset our baselines were trained on. Given the 2 seconds observation trajectory, the goal is to predict the next 3 seconds as the future motion of the vehicle. It is a large scale vehicle trajectory dataset. The dataset covers parts of Pittsburgh and Miami with total size of 290 kilometers of lanes.

4.1.2 Metrics

Hard Off-road Rate (**HOR**): in order to measure the percentage of samples with an inadmissible prediction with regards to the scene, we define HOR as the percentage of scenarios that at least one off-road happens in the prediction trajectory points. It is rounded to the nearest integer.

Soft Off-road Rate (**SOR**): to measure the performance in each scenario more thoroughly, we measure the percentage of off-road prediction points over all prediction points and the average over all scenarios is reported. The reported values are rounded to the nearest integer.

4.1.3 Implementation details

We set the number of iterations to 60, the friction coefficient μ to 0.7 [11] and b equal to 5 for all experiments. For the choice of the black-box algorithm, as the search space of parameters is small in our case, we opt for the brute-force algorithm. We developed our model using a 32GB V100 NVIDIA GPU.

4.2. Results

We first provide the quantitative results of applying our method to the baselines in Table 1. The last column (All) represents the results of the search method described in Section 3.3. We also reported the performance of considering only one category of scene generation functions in the optimization problem Equation (7) in the other columns of the table. The results indicate a substantial increase in SOR and HOR across all baselines in different categories of the generated scenes. This shows that the generated scenes are difficult for the models to handle. LaneGCN and WIMP have competitive performances, but WIMP run-time is 50 times slower than LaneGCN. Hence, we use LaneGCN to conduct our remaining experiments.

Figure 3 visualizes the performance of the baselines in our generated scenes. We observe that all models are challenged with the generated scenes. More cases are provided in the appendix.

In Table 1, we observe that SOR is less than or equal to 1% for all methods in the original scenes. Our exploration shows that more than 90% of these off-road cases are due to the annotation noise in the drivable area maps of the dataset and the models are almost error-free with respect to the scene. Some figures are provided in the appendix. While this might lead to the conclusion that the models are flawless, results on the generated scenes question this conclusion. We confirm our claim in the next section by retrieving the real-world scenes where the model fails.

Feasibility of a scenario is an important feature for generated scenes. As mentioned in Section 3.3, we added physical constraints to guarantee the physical feasibility of the scenes. Table 1 indicates that MPC as a rule-based model predicts almost without any off-road in the generated scenarios. It approves that the scenes are feasible with the given history trajectory. In order to study the importance of added constraints, we relax the constraints for the generated scenes. We report the performance of the baseline and MPC on the cases where the maximum speed in their h is higher than v_{max}. In Table 2, we observe that without those feasibility-assurance constraints, there are more cases where MPC is unable to follow the road and has $3\times$ more off-road. We conclude that those constraints are necessary to make the scene feasible. We keep the constraints in all of our experiments to generate feasible scenarios.

4.3. Real-world retrieval

So far, we have shown that the generated scenes along with the constraints are feasible/realistic scenes. Next, we want to study the plausibility/existence of the generated scenes. Inspired by image retrieval methods [37], we develop a retrieval method to find similar roads in the real-world. First, we extract data of 4 arbitrary cities (New York, Paris, Hong Kong, and New Mexico) using OSM [1]. Then, $20,000$ random samples of 200×200 meters are collected from each city. Note that it is the same view size as in Argoverse samples. Then, a feature extractor is required to obtain a feature vector for each scene. We used the scene feature extractor of LaneGCN named MapNet to obtain some 128 dimensional feature vectors for each sample. We then use the well-known image retrieval method K-tree algorithm [37]. It first uses K-Means algorithm multiple times to cluster the feature vectors of all scenes into a predefined number of clusters (in our case 1000). Then, given a generated scene as the query, it sorts real scenes based on the similarity with the query scene and retrieves 10 closest scenes to the query. Finally, we test the prediction model in these examples. Some examples are provided in Figure 4. More scenes can be found in the appendix.

4.4. Robustness

Here, we study if we can make the models robust against new generated scenes. To this end, we fine-tune the trained model using a combination of the original training data and the generated examples by our method for 10 epochs.

We report the performance of these models in the generated scenes with different transformation power. Transformation power is determined by $\alpha_2 \times 3000$, $\beta_{12} \times 3000$ and γ_1 for Equation (3), Equation (4), and Equation (5), respectively. It represents the amount of curvature in the scene. Table 3 indicates that without losing the performance in the original accuracy metrics, the fine-tuned model is less vulnerable to the generated scenes by predicting 40% less SOR and 30% less HOR in the Full setting. While the results show improvements in all transformation powers, the gains in extreme cases are higher, *i.e.*, the model can handle them better after fine-tuning.

In Figure 5, the prediction of the original model is compared with the prediction of the robust model. The original model cannot predict without off-road while the fine-tuned model is able to predict reasonable and without any off-road point.

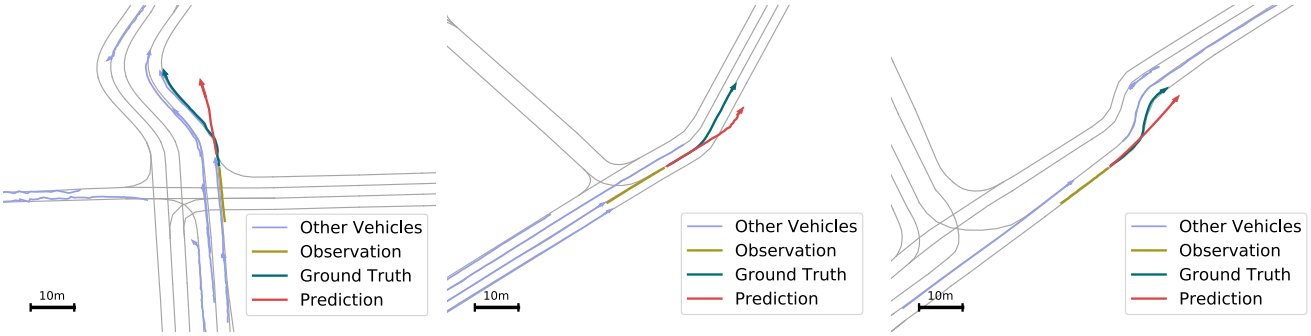
4.5. Discussions

In this section, we perform experiments and bring speculations to shed light on the weaknesses of the models.

1. We study the ability to transfer the generated scenes to new models, *i.e.,* how models perform on the scenes

Model	Original	Generated (**Ours**)			
		Smooth-turn	Double-turn	Ripple-road	All
	SOR / HOR	SOR / HOR	SOR / HOR	SOR / HOR	SOR / HOR
DATF [38]	1 / 2	37 / 77	36 / 76	42 / 80	43 / 82
WIMP [27]	0 / 1	13 / 46	14 / 50	20 / 58	22 / 63
LaneGCN [32]	0 / 1	8 / 40	19 / 60	21 / 62	23 / 66
MPC [56]	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0

Table 1. **Comparing the performance of different baselines in the original dataset scenes and our generated scenes.** SOR and HOR are reported in percent and the lower represent a better reasoning on the scenes by the model. MPC as a rule-based model always has on-road predictions both in original and our generated scenes.



(a) DATF (b) WIMP (c) LaneGCN

Figure 3. **The predictions of different models in some generated scenes.** All models are challenged by the generated scenes and failed in predicting in the drivable area.

Model	w/ phys	w/o phys
	SOR / HOR	SOR / HOR
LaneGCN	33 / 85	47 / 92
MPC	0 / 1	0 / 3

Table 2. **Impact of the physical constraints.** We report the performance with and without the physical constraints explained in Section 3.3. The numbers are reported on samples of data with speed higher than v_{max} in their h.

generated for other models. We conduct this experiment by storing the generated scenes for a source model which lead to an off-road prediction, and evaluate the performance of target models on the stored scenes. Table 4 shows that the transferred scenes are still difficult cases for other models.

2. We study how models perform with smoothly changing the transformation functions parameters. To this end, we smoothly change the transformation parameters for 100 random scenes and visualize the heatmap of HOR for the generated scenes. Figure 6 demonstrates that models are more vulnerable to larger transformation parameters, *i.e.*, sharper turns. Also, it shows more off-road in the left turns compared with the right ones which could be due to the biases in the dataset [36]. A clear improvement is visible in the robust model.

3. Our experiments showed that while the model has almost zero off-road rate in the original scenes, it suffers from over 60% off-road rate in the generated ones. In order to hypothesize the causes of this gap, we explored the training data. We observed that in most samples, the history h has enough information about the future trajectory which reduces the need for the scene reasoning. However, our scene generation approach changes the scene such that h includes almost no information about the future trajectory. This essentially makes a situation which requires scene reasoning. We speculate that this feature is one factor which makes the generated scenes challenging. Note that this does not contradict with the ablations in [32] as their performance measure is accuracy. Figure 7a shows a failure of the model where the prediction is only based on h instead of reasoning over the scene. However, the robust model learned to reason over the scene, as shown in Figure 7b. While our discussion is an observational

(a) Paris location (b) Hong Kong location (c) New Mexico location

Figure 4. **Retrieving some real-world locations similar to the generated scenes using our real-world retrieval algorithm.** We observe that the model fails in Paris (a), Hong Kong (b) and New Mexico (c).

Model	Pow=1 SOR/HOR	Pow=3 SOR/HOR	Pow=5 SOR/HOR	Pow=7 SOR/HOR	Pow=9 (Full) SOR/HOR
LaneGCN	2 / 8	12 / 35	19 / 49	22 / 58	23 / 66
LaneGCN w/ aug	**1 / 7**	**6 / 21**	**10 / 30**	**13 / 38**	**14 / 46**

Table 3. **Comparing the original model and the fine-tuned model with data augmentation of the generated scenes.** The performance is reported on generated scenes with different transformation power (Pow). Transformation power is determined by $\alpha_2 \times 3,000$, $\beta_{12} \times 3,000$ and γ_1 for Equation (3), Equation (4), and Equation (5), respectively which represents the amount of curvature in the scene. The average / final displacement errors on original scenes are equal to $1.35/2.98m$ for both original and fine-tuned models.

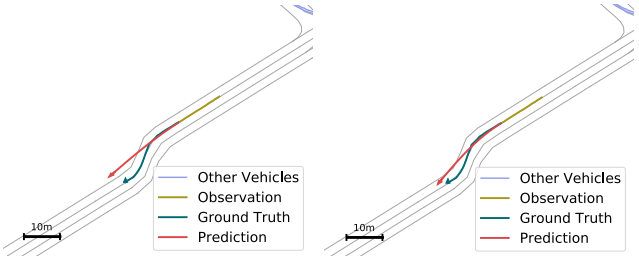


Figure 5. **The output of the original model (the left) vs the robust model (the right) in a generated scene.** While the original model has a trajectory in non-drivable area, the robust model predicts without any off-road.

Source models	Target models		
	LaneGCN	DATF	WIMP
LaneGCN	34 / 100	37 / 82	20 / 61
DATF	11 / 44	52 / 100	13 / 46
WIMP	20 / 63	40 / 82	36 / 100

Table 4. **Studying the transferability of the generated scenes.** We generate scenes for source model and keep the ones that have off-road prediction by the source model. The target models are evaluated using those scenes. The reported numbers are SOR/HOR values. Numbers are rounded to the nearest integer.

hypothesis, we leave further studies for future works.

4. In some cases, our generated scene could not lead to an off-road prediction. One such example is depicted in Figure 8a.

5. While our method offers a new approach for assessing trajectory prediction models, it has some limitations. First, our transformation functions are limited, and they cannot cover all real-world cases. We however propose a general methodology that can be expanded by adding other types of transformations. To demonstrate it, we add lane merging to the framework, which causes 14% HOR. Second, in addition to the off-road criterion, there exist other failure criteria. For instance, collision with other agents or abnormal behaviors like sudden lane changes. By choosing collision with other agents as criterion, HOR becomes 1.68% in the generated scenes while it is 0.55% in original data.

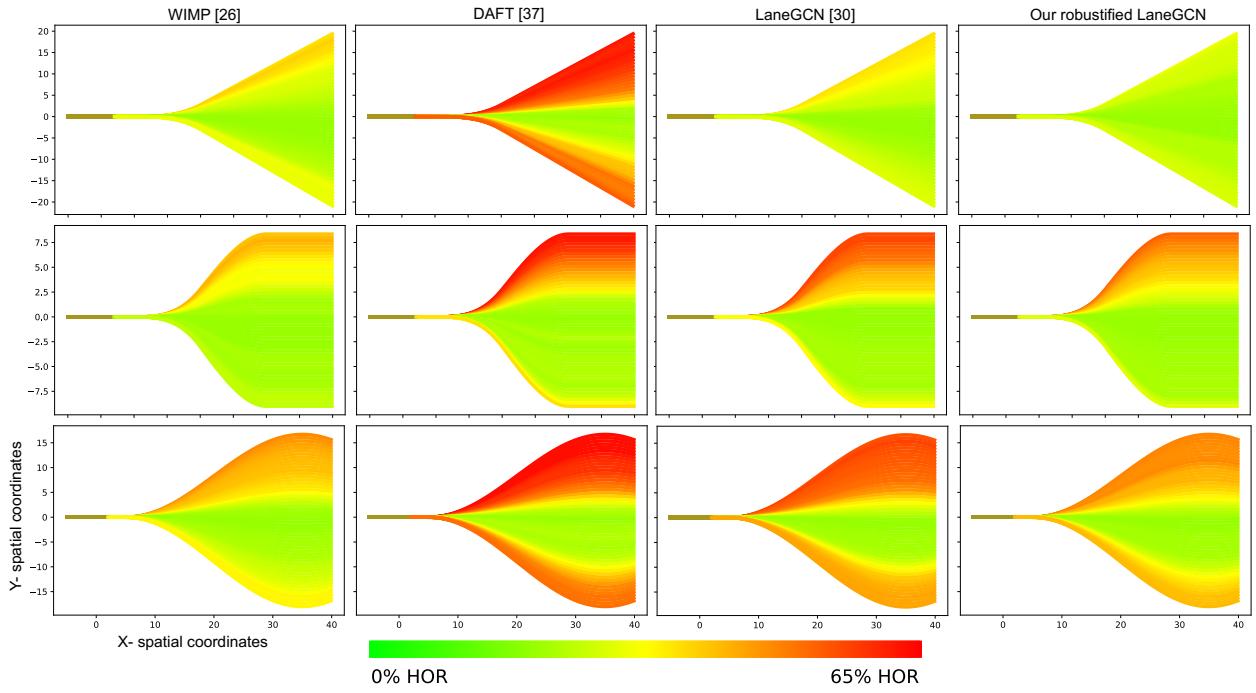
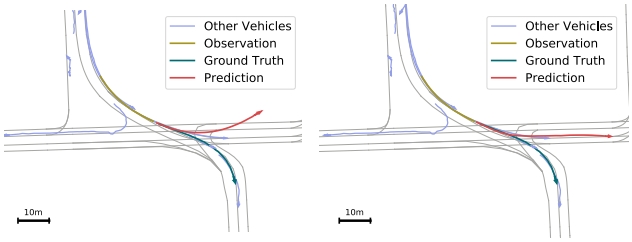
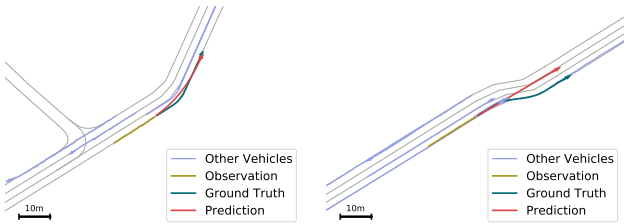


Figure 6. **The qualitative results of baselines for different transformation functions.** The red color indicates more off-road prediction in those scenes and the green indicates higher admissible ones. Usually the models fail in turns with high curvature. We could successfully make the LaneGCN model more robust by fine-tuning.



(a) Before robustness (b) After robustness

Figure 7. **The output of the model before and after the robustness in a sample which requires reasoning over the scene.** We observe that the model before robustness mainly uses h to predict instead of reasoning over the scene. However, after robustness, it reasons more over the scene.



(a) (b)

Figure 8. **Some successful cases of the prediction model.** In (a), the model follows the road and predicts without any off-road. In (b), while the model predicts on-road, it suddenly changes its lane.

Moreover, Figure 8b shows one scenario in which the predictions of the model are in the drivable area but the sudden lane change is abnormal.

5. Conclusion

In this work, we presented a conditional scene generation method. We showed that several state-of-the-art trajectory prediction models fail in our generated scenes. Notably, they have high off-road rate in their predictions. Next, leveraging image retrieval techniques, we retrieved real-world locations which partially resemble the generated scenes and demonstrate their failure in those locations. We made the model robust against the generated scenes. We hope that this framework helps to better evaluate the prediction models which are involved in the autonomous driving systems.

6. Acknowledgments

This project was funded by Honda R&D Co., Ltd and from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 754354. We thank Lorenzo Bertoni and Kirell Benzi for their valuable feedback.

References

[1] Openstreetmap. https://www.openstreetmap.org. 2, 5

[2] Argoai challenge. *CVPR Workshop on Autonomous Driving*, 2020. 4

[3] Yasasa Abeysirigoonawardena, Florian Shkurti, and Gregory Dudek. Generating adversarial driving scenarios in high-fidelity simulators. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2019. 2

[4] Matthias Althoff and Sebastian Lutz. Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles. In *IEEE Intelligent Vehicles Symposium (IV)*, 2018. 2

[5] Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, Huan Zhang, Cho-Jui Hsieh, and Mani B. Srivastava. Genattack: practical black-box attacks with gradient-free optimization. *GECCO*, 2019. 4

[6] Mohammadhossein Bahari, Ismail Nejjar, and Alexandre Alahi. Injecting knowledge in data-driven vehicle trajectory predictors. *Transportation Research Part C: Emerging Technologies*, 128, 2021. 4

[7] Mohammadhossein Bahari, Vahid Zehtab, Sadegh Khorasani, Sana Ayramlou, Saeed Saadatnejad, and Alexandre Alahi. Svg-net: An svg-based trajectory prediction model. *arXiv preprint arXiv:2110.03706*, 2021. 2

[8] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *Robotics: Science and Systems (RSS)*, 2019. 1, 2

[9] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems(NeurIPS)*, 2011. 4

[10] Yuriy Biktairov, Maxim Stebelev, Irina Rudenko, Oleh Shliazhko, and Boris Yangel. Prank: motion prediction based on ranking. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[11] Peter J. Blau. *Friction Science and Technology: From Concepts to Applications*. CRC Press, 2005. 5

[12] Adith Boloor, Karthik Garimella, Xin He, Christopher Gill, Yevgeniy Vorobeychik, and Xuan Zhang. Attacking vision-based perception in end-to-end autonomous driving models. *Journal of Systems Architecture*, 110, 2020. 2

[13] Smail Ait Bouhsain, Saeed Saadatnejad, and Alexandre Alahi. Pedestrian intention prediction: A multi-task perspective. *European Association for Research in Transportation (hEART)*, 2020. 2

[14] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[15] Yulong Cao, Chaowei Xiao, Dawei Yang, Jing Fang, Ruigang Yang, Mingyan Liu, and Bo Li. Adversarial objects against lidar-based autonomous driving systems. *arXiv preprint arXiv:1907.05418*, 2019. 2

[16] Sergio Casas, Cole Gulino, Simon Suo, and Raquel Urtasun. The importance of prior knowledge in precise multimodal prediction, 2020. 1, 2

[17] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Conference on Robot Learning*. PMLR, 2018. 2

[18] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *Conference on Robot Learning*, 2019. 2

[19] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 4

[20] Anthony Corso, Robert J Moss, Mark Koren, Ritchie Lee, and Mykel J Kochenderfer. A survey of algorithms for black-box safety validation. *arXiv preprint arXiv:2005.02979*, 2020. 2

[21] Pasquale Coscia, Francesco Castaldo, Francesco A.N. Palmieri, Alexandre Alahi, Silvio Savarese, and Lamberto Ballan. Long-term path prediction in urban scenarios using circular distributions. *Image and Vision Computing*, 69, 2018. 2

[22] Halliday David, Robert Resnick, and Jearl Walker. *Fundamentals of physics*. Wiley, 1997. 3

[23] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1

[24] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[25] Joey Hong, Benjamin Sapp, and James Philbin. Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[26] Francis Indaheng, Edward Kim, Kesav Viswanadha, Jay Shenoy, Jinkyu Kim, Daniel J Fremont, and Sanjit A Seshia. A scenario-based platform for testing autonomous vehicle behavior prediction models in simulation. *arXiv preprint arXiv:2110.14870*, 2021. 2

[27] Siddhesh Khandelwal, William Qi, Jagjeet Singh, Andrew Hartnett, and Deva Ramanan. What-if motion prediction for autonomous driving. *arXiv preprint arXiv:2008.10587*, 2020. 4, 6

[28] Moritz Klischat and Matthias Althoff. Generating critical test scenarios for automated vehicles with evolutionary algorithms. In *IEEE Intelligent Vehicles Symposium (IV)*, 2019. 2

[29] Zelun Kong, Junfeng Guo, Ang Li, and Cong Liu. Physgan: Generating physical-world-resilient adversarial examples for

autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[30] Parth Kothari, Sven Kreiss, and Alexandre Alahi. Human trajectory forecasting in crowds: A deep learning perspective. *IEEE Transactions on Intelligent Transportation Systems*, 2021. 2

[31] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[32] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *European Conference on Computer Vision (ECCV)*. Springer, 2020. 1, 2, 4, 6

[33] Harshitha Machiraju and Vineeth N Balasubramanian. A little fog for a large turn. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020. 2

[34] Kim-Fung Man, Wallace Kit-Sang Tang, and Sam Kwong. Genetic algorithms: concepts and applications [in engineering design]. *IEEE Trans. Ind. Electron.*, 1996. 4

[35] Lu Mi, Hang Zhao, Charlie Nash, Xiaohan Jin, Jiyang Gao, Chen Sun, Cordelia Schmid, Nir Shavit, Yuning Chai, and Dragomir Anguelov. Hdmapgen: A hierarchical graph generative model of high definition maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[36] Matthew Niedoba, Henggang Cui, Kevin Luo, Darshan Hegde, Fang-Chieh Chou, and Nemanja Djuric. Improving movement prediction of traffic actors using off-road loss and bias mitigation. *Advances in Neural Information Processing Systems (NeurIPS) Workshop*, 2019. 1, 6

[37] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2006. 5

[38] Seong Hyeon Park, Gyubok Lee, Jimin Seo, Manoj Bhat, Minseok Kang, Jonathan Francis, Ashwin Jadhav, Paul Pu Liang, and Louis-Philippe Morency. Diverse and admissible trajectory forecasting through multimodal context understanding. In *European Conference on Computer Vision (ECCV)*. Springer, 2020. 1, 2, 4, 6

[39] Behnam Parsaeifard, Saeed Saadatnejad, Yuejiang Liu, Taylor Mordan, and Alexandre Alahi. Learning decoupled representations for human pose forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) workshop*, pages 2294–2303, 2021. 2

[40] Xuanchi Ren, Tao Yang, Li Erran Li, Alexandre Alahi, and Qifeng Chen. Safety-aware motion prediction with unseen vehicles for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[41] Nicholas Rhinehart, Kris M. Kitani, and Paul Vernaza. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2

[42] Binxin Ru, Adam D. Cobb, Arno Blaas, and Yarin Gal. Bayesopt adversarial attack. *International Conference on Learning Representations (ICLR)*, 2020. 4

[43] Saeed Saadatnejad, Mohammadhossein Bahari, Pedram Khorsandi, Mohammad Saneian, Seyed-Mohsen Moosavi-Dezfooli, and Alexandre Alahi. Are socially-aware trajectory prediction models really socially-aware? *arXiv preprint arXiv:2108.10879*, 2021. 2

[44] Saeed Saadatnejad, Siyuan Li, Taylor Mordan, and Alexandre Alahi. A shared representation for photorealistic driving simulators. *IEEE Transactions on Intelligent Transportation Systems*, 2021. 2

[45] Amir Sadeghian, Ferdinand Legros, Maxime Voisin, Ricky Vesel, Alexandre Alahi, and Silvio Savarese. Car-net: Clairvoyant attentive recurrent network. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[46] Takami Sato, Junjie Shen, Ningfei Wang, Yunhan Jia, Xue Lin, and Qi Alfred Chen. Dirty road can attack: Security of deep learning based automated lane centering under physical-world attack. In *Proceedings of the 29th USENIX Security Symposium (USENIX Security'21)*, 2021. 2

[47] Haoran Song, Di Luan, Wenchao Ding, Michael Yu Wang, and Qifeng Chen. Learning to predict vehicle trajectories with model-based planning. In *Conference on Robot Learning*, 2021. 2

[48] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *International Conference on Machine Learning (ICML)*, 2010. 4

[49] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z Morley Mao. Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th USENIX Security Symposium*, 2020. 2

[50] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th international conference on software engineering*, 2018. 2

[51] James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. Physically realizable adversarial examples for lidar object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[52] Akifumi Wachi. Failure-scenario maker for rule-based agent using multi-agent adversarial reinforcement learning and its application to autonomous driving. *Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, 2019. 2

[53] Jingkang Wang, Ava Pun, James Tu, Sivabalan Manivasagam, Abbas Sadat, Sergio Casas, Mengye Ren, and Raquel Urtasun. Advsim: generating safety-critical scenarios for self-driving vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[54] Mengshi Zhang, Yuqun Zhang, Lingming Zhang, Cong Liu, and Sarfraz Khurshid. Deeproad: Gan-based metamorphic

testing and input validation framework for autonomous driving systems. In *33rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2018. 2

[55] Husheng Zhou, Wei Li, Zelun Kong, Junfeng Guo, Yuqun Zhang, Bei Yu, Lingming Zhang, and Cong Liu. Deepbillboard: Systematic physical-world testing of autonomous driving systems. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, 2020. 2

[56] Julius Ziegler, Philipp Bender, Markus Schreiber, Henning Lategahn, Tobias Strauß, Christoph Stiller, Thao Dang, Uwe Franke, Nils Appenrodt, Christoph Gustav Keller, Eberhard Kaus, Ralf G. Herrtwich, Clemens Rabe, David Pfeiffer, Frank Lindner, Fridtjof Stein, Friedrich Erbs, Markus Enzweiler, Carsten Knöppel, Jochen Hipp, Martin Haueis, Maximilian Trepte, Carsten Brenk, Andreas Tamke, Mohammad Ghanaat, Markus Braun, Armin Joos, Hans Fritz, Horst Mock, Martin Hein, and Eberhard Zeeb. Making bertha drive—an autonomous journey on a historic route. *IEEE Intelligent Transportation Systems Magazine*, 6, 2014. 4, 6