# On Adversarial Robustness of Trajectory Prediction for Autonomous Vehicles: Supplementary Materials

Qingzhao Zhang[1], Shengtuo Hu[1], Jiachen Sun[1], Qi Alfred Chen[2], Z. Morley Mao[1]
[1]University of Michigan, [2]University of California, Irvine
{qzzhang, shengtuo, jiachens, zmao}@umich.edu alfchen@uci.edu

Table 1. Maximum bounds of physical properties.

| Dataset | $\|v\|$ | $a_l$ | $a_s$ | $da_l/dt$ | $da_s/dt$ |
|---|---|---|---|---|---|
| Apolloscape [4] | 21.078 | 9.914 | 1.912 | 16.836 | 3.154 |
| NGSIM [1] | 20.830 | 1.455 | 0.620 | 1.955 | 0.925 |
| nuScenes [3] | 17.198 | 2.550 | 0.936 | 3.914 | 1.070 |

**Notes**: $|v|$ – scalar velocity ($m/s$); $a_l$ – longitudinal acceleration ($m/s^2$); $a_s$ – lateral acceleration ($m/s^2$); $da_l/dt$ – derivative of longitudinal acceleration ($m/s^3$); $da_s/dt$ – derivative of lateral acceleration ($m/s^3$).

# 1. Implementation Details

## 1.1. Open-sourced Code

Our implementation is open source at Github (`https://github.com/zqzqz/AdvTrajectoryPrediction`), which is a framework of testing adversarial attacks and mitigation methods on trajectory prediction algorithms.

## 1.2. Bounds of Physical Properties

We restrict 5 physical properties of perturbed trajectories: (1) scalar velocity, (2) longitudinal acceleration, (3) lateral acceleration, (4) derivative of longitudinal acceleration, and (5) derivative of lateral acceleration. For the three datasets (i.e., Apolloscape, NGSIM, and nuScenes), we select different bounds of physical properties according to the data distribution following the approach mentioned in the main paper. We report the values of such bounds in Table 1.

## 1.3. Hyper-parameters of Attacks

In general, both white-box and black-box attacks are iteration-based methods, which require parameters about evolution speed and maximum iterations. When tuning the parameters, we monitor the objective loss over time. The parameters are proper if the loss is overall decreasing and stays low stably in the end.

For the PGD-based white box attack, we use Adam optimizer with a learning rate of 0.01 and set the maximum iteration to 100. For PSO-based black box attack, we set the number of particles to 10, inertia weight to 1.0, acceleration coefficients to (0.5, 0.3), and the maximum iteration to 100.

## 1.4. Implementation of Mitigation

For data augmentation, we randomly select 50% (the parameter is fine-tuned) of trajectories to add perturbation during the training. The added perturbation is random and under the hard constraints of perturbation. We also double the maximum training iteration.

For trajectory smoothing, we use a linear smoother with a convolution kernel $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, which takes the mean of three trajectory positions as the smoothed position at the middle time frame. Formally, we denote the trajectory as $s_1, \ldots, s_i, \ldots, s_N$ where $N$ is the total length of the trajectory and $s_i$ ($i \in 1 \ldots N$) is the two-dimensional trajectory location at time frame $i$. The smoothed trajectory locations $s'_i = \frac{1}{3}s_{i-1} + \frac{1}{3}s_i + \frac{1}{3}s_{i+1}$ ($i \in 1 \ldots N-1$).

For detection, we prepare a set of normal trajectories (other than training/testing data used in the attack experiments) and generate a set of abnormal trajectories by adding random perturbation on normal trajectories. We use the normal and perturbed trajectories together to fit the SVM model and find out proper thresholds. The SVM model uses RBF kernel, implemented in the *scikit-learn* library. The threshold parameters for detection are selected to achieve the best Area Under Curve (AUC) score of Receiver Operating Characteristics (ROC) curve.

# 2. Hard Scenarios for Prediction

In the main paper we mention that existing prediction models have high prediction error on several hard scenarios. In this section, we visualize two scenarios in Figure 1 and Figure 2. In each figure, we present the ground-truth trajectory of the target vehicle and prediction made by tested models. In Figure 1, the target vehicle takes brake in its future trajectory and finally stops at the stop sign. However, the prediction models cannot foresee such braking behavior and the predicted trajectory still has a steady velocity. In Figure 2, the target vehicle turns right in its future trajectory but the pattern of turning is not obvious in the history trajec-

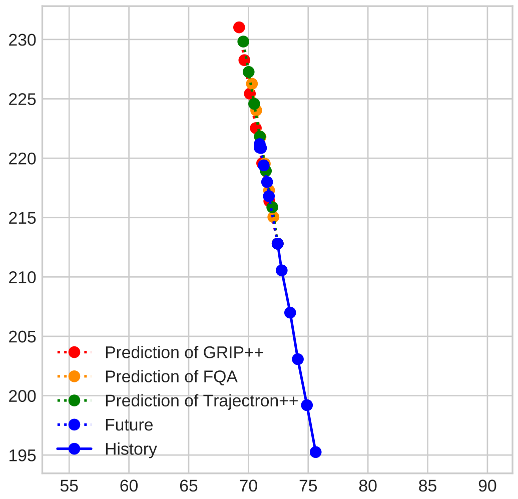


Figure 1. Hard scenario: stopping at a stop sign.

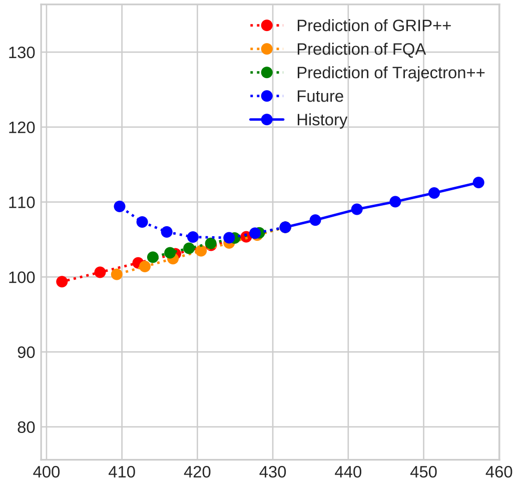


Figure 2. Hard scenario: turning right at an intersection.

tory. Therefore, it is hard for prediction models to correctly predict such turning trajectory.

## 3. More Case Studies

The case study in the main paper discusses a scenario where the attacker leverages high prediction error to spoof a fake lane changing behavior. In this section, we show two more case studies to demonstrate attack impacts considering various attack goals.

First, the adversarial trajectories with high prediction error can also hides existing driving behaviors instead of spoofing behaviors. As shown in Figure 3, the OV is in fact shifting to the left lane (the AV's lane) and the prediction correctly captures the behavior without perturbation. The average deviation to right is 0.61 meter originally. Under the white-box attack (3-second length, 1-meter deviation bound, maximizing deviation to right), the adversarial trajectory is still natural but the predicted trajectories are going straight without any pattern of changing lanes. The average deviation to right is increased to 1.22 meters (2×). The attack significantly reduces the time for the AV to safely respond to the lane changing behavior. Originally, the AV learn the OV will change lane at time frame 1 because the prediction at time frame 2 already cross with the

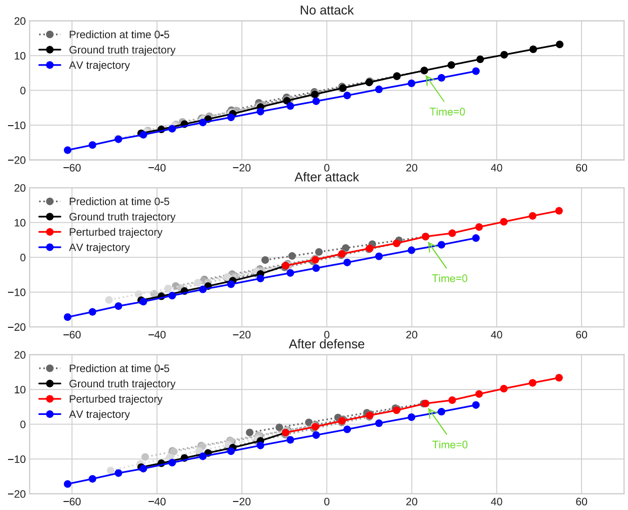


Figure 3. Case study: deviation to right hides a real lane changing behavior. GRIP++ model, Apolloscape dataset, white-box multi-frame attack, mitigation of data augmentation and train-time smoothing.

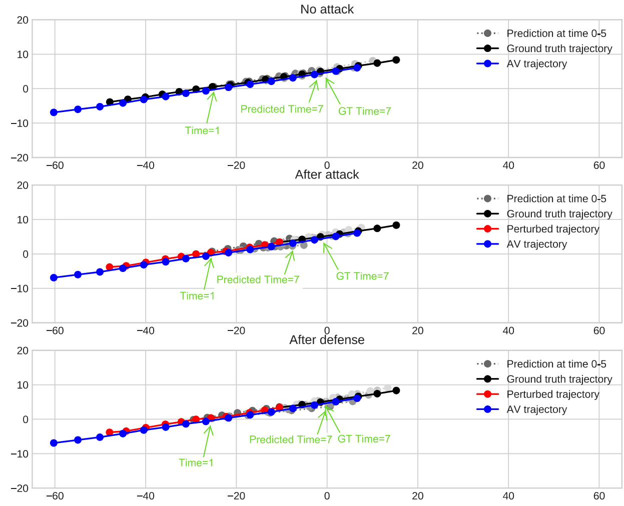


Figure 4. Case study: deviation to rear spoofs a fake braking behavior. GRIP++ model, Apolloscape dataset, white-box multi-frame attack, mitigation of data augmentation and train-time smoothing.

AV's future trajectory. After the attack, the AV does not acknowledge the lane changing until time frame 7. Hence, the AV receives the lane changing signal 2.5 seconds late which may cause a hard brake or even a rear-end collision. What is worse, the mitigation is not effective on this case because the adversarial trajectory is smooth and natural between time frame 1 to 6.

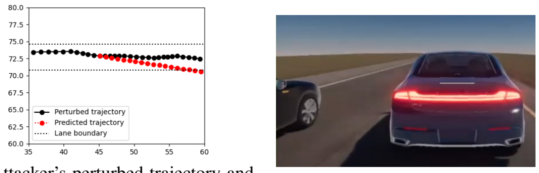
Second, the longitude deviation is as dangerous as the lateral deviation. In Figure 4, the OV is driving in front of the AV with a distance of about 10 meters. Originally, the OV moves in a almost constant velocity and the average deviation to rear direction is 0.22 meter. After the white-box attack (3-second length, 1-meter deviation bound, maximizing deviation to rear), the OV stays in its route but the velocity is not stable. On the adversarial trajectory, the predicted trajectories show that OV is going to decelerate and the deviation to rear is increased to 2.99 meters (14×). In time frame 2 for instance, the length of the predicted trajectory is shortened by 50% and the AV's planning logic decides to deceleration to avoid a potential collision. Depending on the distance between the OV and the AV, the attack may cause speed drop, a hard brake, or collision of the AV. Mitigation of trajectory smoothing is effective on this case. The trajectory smoothing alleviates the fluctuation of the OV's velocity.

## 4. Realistic Attack Setting

As discussed in the threat model, the adversary does not have the ground truth of other vehicles/pedestrians when computing the adversarial trajectory. Therefore, we consider a more realistic setting where the adversary includes predicted trajectories in the input of generating adversarial examples. We select 75 scenarios in *Apolloscape* dataset and analyze attacks on *GRIP++* model. In each scenario, the adversary has full knowledge of trajectories in the first 3 seconds, predicts trajectories of all other objects between 3-6 seconds using *GRIP++*, and generates the adversary trajectory between 3-6 seconds. Prediction error (six metrics) of the above attack is 6.82/10.41/2.61/2.38/3.92/5.53 meters, which is 95% of the attacks using ground-truth object trajectories. The adversary can approximate the future knowledge to generate attacks whose effectiveness is almost equal to ideal white-box attacks in Table 3 in the main paper because the target vehicle's trajectory itself dominates the prediction results.

## 5. Attack Real-world AV System

Our attack is effective on real AV software, Baidu Apollo 6.0 [2], which uses an LSTM predictor on 2 seconds of history trajectory in a frequency of 10 Hz. Figure 5 shows that the adversarial trajectory spoofs fake lane changing (in the left figure), resulting in a brake of the right Apollo AV (in LGSVL simulator [5]). Second, history length and frequency are system-specific settings, and prediction models mostly choose 2-3 seconds of history and 2-10 Hz. Although we showed different values for the two parameters in Tab.1, we did not observe a clear correlation between the parameters and prediction accuracy. We can openly discuss this question as future work.



(a) Attacker's perturbed trajectory and prediction made by Apollo.

(b) Simulator LGSVL's view.

Figure 5. Reproducing a prediction attack on Baidu Apollo 6.0.

## References

[1] Traffic Analysis Tools: Next Generation Simulation. https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm, 2020. 1

[2] Baidu Apollo. http://apollo.auto, 2021. 3

[3] 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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 1

[4] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 954–960, 2018. 1

[5] Guodong Rong, Byung Hyun Shin, Hadi Tabatabaee, Qiang Lu, Steve Lemke, Mārtiņš Možeiko, Eric Boise, Geehoon Uhm, Mark Gerow, Shalin Mehta, et al. Lgsvl simulator: A high fidelity simulator for autonomous driving. In *2020 IEEE 23rd International conference on intelligent transportation systems (ITSC)*, pages 1–6. IEEE, 2020. 3