Fooling LiDAR Perception via Adversarial Trajectory Perturbation
Supplementary Materials

Yiming Li[†,*] Congcong Wen[†,§,*] Felix Juefei-Xu[‡] Chen Feng[†]
[†]New York University [§]University of Chinese Academy of Sciences [‡]Alibaba Group
yimingli@nyu.edu, cfeng@nyu.edu

I. Overview

In the supplementary, we first present more results of our FLAT attack against black box PointPillar++ [11]. Then more qualitative evaluations of white box attack are shown, and the performances under different parameters (the number of attack iteration, the step size for each iteration, and the interpolation step) are discussed. Finally, more examples of ***polynomial trajectory perturbation*** (including both the point cloud after our attack and and the trajectory perturbation) are presented to validate the imperceptibility in point cloud space and the strong smoothness in trajectory.

II. Black Box Attack

II.1. Qualitative Evaluation

Additional qualitative examples[1] are presented in Fig. I and Fig. II. Although our FLAT attack cannot know the model parameters of PointPillar++, a subtle perturbation can make the detector lose many safety-critical objects, *e.g.*, as for the four sweeps in Fig. I, the adversarial perturbation in the full trajectory can make the detector lose 7, 20, 8 and 6 objects respectively (from top to bottom), severely damaging the self-driving car's perception module. Moreover, false positives are also increased by our perturbation (6 more in the third row of Fig. I), making the car mistakenly believe that there are obstacles in the free space.

II.2. Quantitative Evaluation

The nuScenes dataset [2] employs 2D center distance (0.5, 1, 2, 4 meters) as the matching threshold when calculating the Average Precision (AP). The per category precision-recall plots of the original detector as well as four attack settings are shown from Fig. III - Fig. VII. The performance drop is the largest when the threshold is 0.5m (the highest precision standard), *e.g.*, the AP@0.5m in car category is decreased by 23.1(33.1%), 36.5(52.4%), 41.5(59.6%) while attacking translation, rotation and the full trajectory. As shown in Fig. VIII, the curve shifts to the lower left when increasing the attack magnitude.

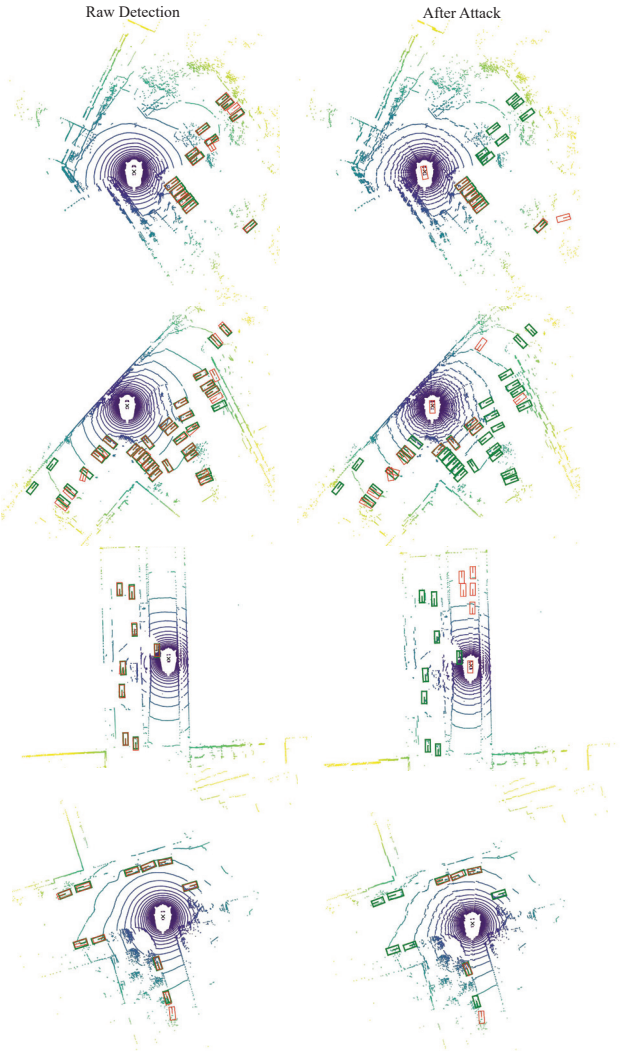


Figure I. Qualitative results of black box attack when attacking the full trajectory. The left and right figures are respectively original and distorted LiDAR sweep as well as the detection results. **Green**/**red** boxes denote the ground truth/prediction respectively.

[1]Noted that we transfer the adversarial perturbation of the white-box *full trajectory* attack targeting *classification of stage-2*

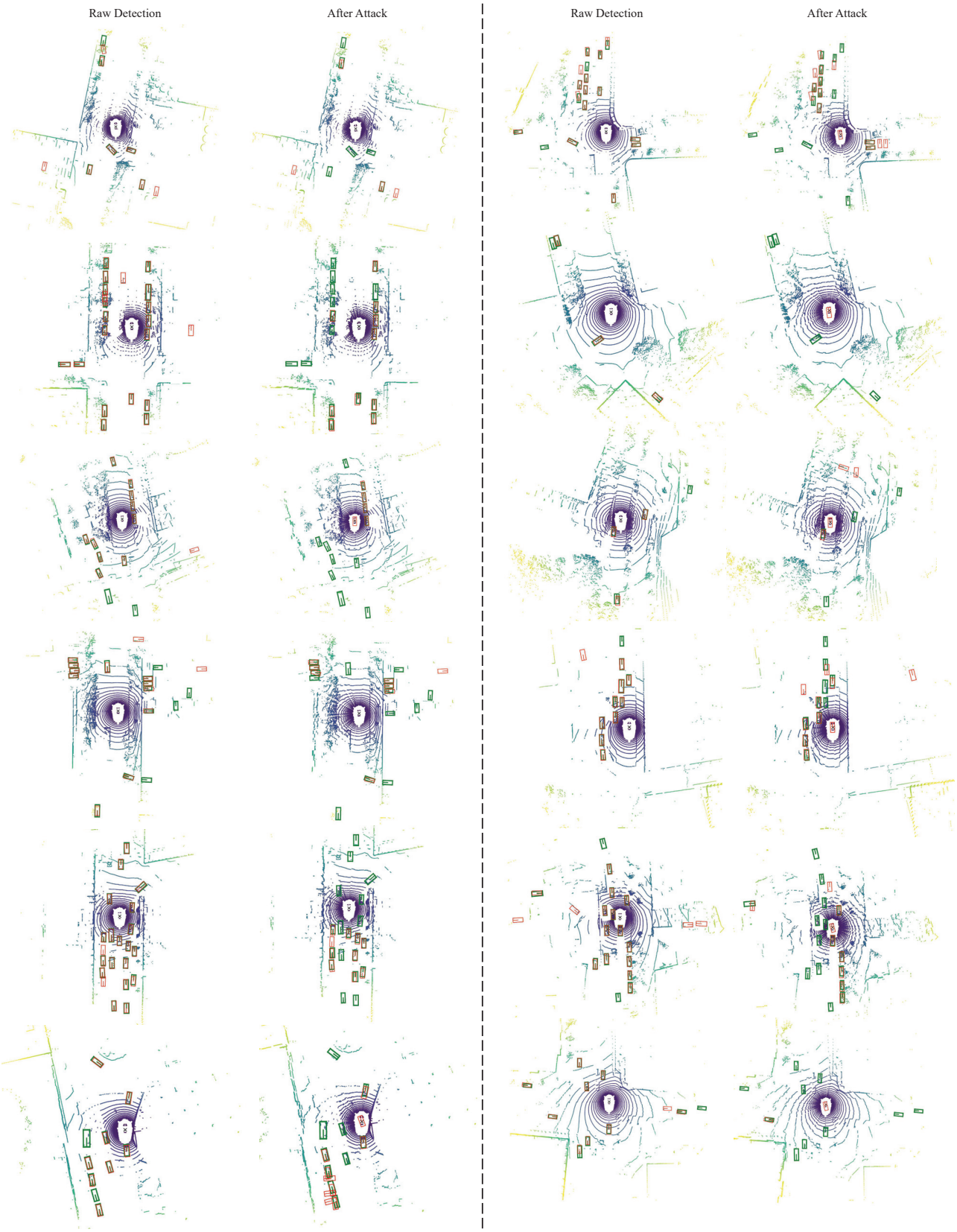
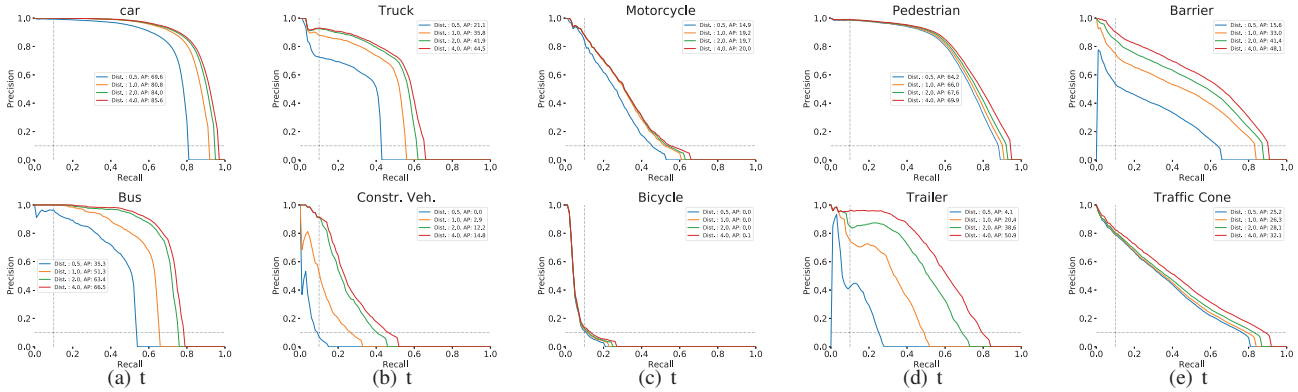
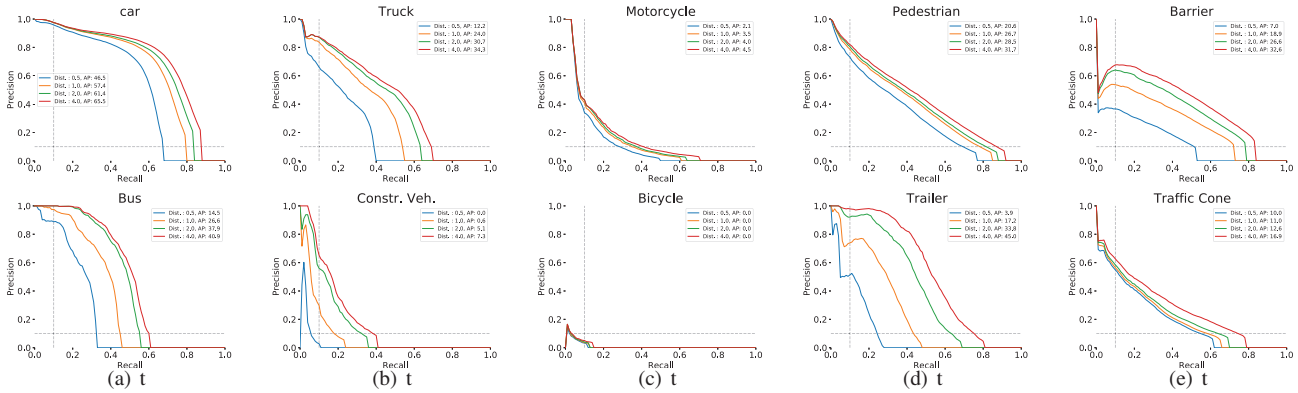


Figure II. Qualitative results of the black box attack when attacking the full trajectory. **Green**/**red** boxes denote the ground truth/prediction respectively.



(a) t (b) t (c) t (d) t (e) t

Figure III. Per category precision-recall plot of **PointPillar++** on the nuScenes validation set [2].



(a) t (b) t (c) t (d) t (e) t

Figure IV. Per category precision-recall plot of attacking the **translation** only (black box) on the nuScenes validation set [2].

III. White Box Attack

III.1. Qualitative Evaluation

Three qualitative examples are demonstrated in Fig. IX - Fig. XI. In Fig. IX, original PointRCNN can perceive three our of five cars, and attacking translation only has slightly shifted the predictions. Attacking rotation has a stronger impact and cause two objects to be undetectable, while attacking the full trajectory further creates four false positives. In Fig. X, PointRCNN can successfully detect two out of three surrounding cars, then adversarial translation perturbation make the detection of the nearest car drift. Both two detections are drifted in the scenario of adversarial rotation perturbation. Finally, when attacking both translation and rotation, five false positives have emerged, severely degrading the perception capability of the self-driving car. Similar results can be found in Fig. XI. Moreover, three qualitative examples of attacking with and without regularization are presented in Fig. XII, Fig. XIII and Fig. XIV. As the strength of regularization is enlarged, the variation in the point space is reduced, enabling less perceptible attack.

III.2. Different Parameters

The detection performances under different parameter settings are reported in Table I - Table III. In the case of attacking the translation (fooling classification in stage-2), AP of easy case in six parameter settings fluctuate between 11.72 and 13.89, which means that all the parameter configurations can produce a high-quality adversarial perturbation. In the situation of attacking the full trajectory (with the aim of attacking regression branch in stage-1), AP of easy case in six parameter settings fluctuate between 0.53 and 2.44, proving that all the six settings can maintain the attacking quality at a high level. To sum up, the number of iteration and the step size for each iteration has a relatively small impact on our FLAT attack. As for the interpolation step, when it is increased, there will be more sectors, *i.e.*, more point cloud groups, and the number of trajectory points which can be attacked are also increased, easing the learning of an effective adversarial perturbation. Consequently, when the interpolation step is increased from 50 to 1000, the AP in easy scenarios can be lowered by 5.04 (36.7%) while attacking the translation, by 3.70 (78.2%)

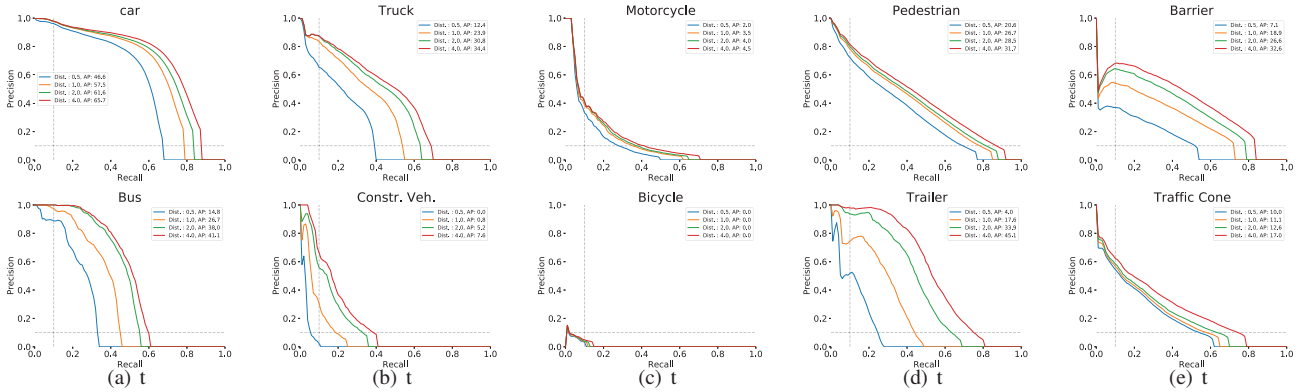


Figure V. Per category precision-recall plot of attacking the **polynomial coefficients** (black box) on the nuScenes validation set [2].

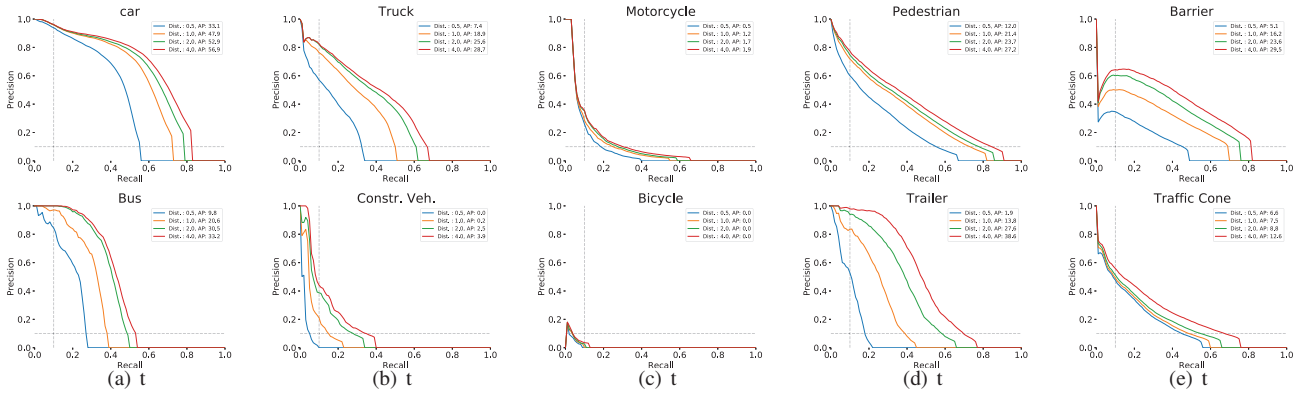


Figure VI. Per category precision-recall plot of attacking the **rotation** only (black box) on the nuScenes validation set [2].

while attacking the rotation, and by 0.51 (87.9%) while attacking the full trajectory, as shown in Table IV.

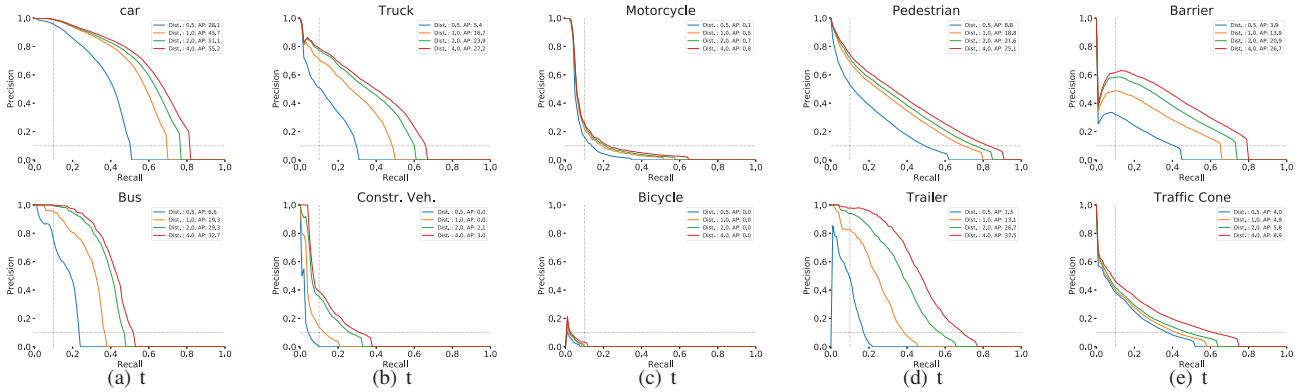
Noted that our simulation of motion distortion/correction accurately matches with the real-world scenarios, *e.g.*, in Fig. 2 of [29]: "the Velodyne VLP-16 software used in our electric vehicle platform produces 76 packets for each full revolution scan. Each packet covers an azimuth angle of approximately 4.74°." Noted that the number of point cloud sector/packet is adjustable depending on vehicle platforms, and the attack performance keeps at a high level under different number of sector/interpolation step, as shown in Table IV.

IV. Polynomial Trajectory Perturbation

Our trajectory attack is feasible, *e.g.*, through GNSS spoofing as proven in [23]: "Today, it is feasible to execute GNSS spoofing attacks with less than $100 of equipment. GNSS signal generators can be programmed to transmit radio frequency signals corresponding to a static position, or simulate entire trajectories." We have verified the effectiveness of the discrete adversarial trajectory perturbation in both white box and black box attack. To achieve a temporally-smooth attack which is less perceptible, we implement a polynomial regression before the generation of perturbation and attack the polynomial coefficients instead of the trajectory itself. In this scenario, we only need to manipulate several key points to bend a polynomial-parameterized trajectory which can be easily achieved in reality, realizing a real-time and high-quality attack. Several qualitative examples of the polynomial trajectory perturbation are shown in Fig. XV. Although the attack performance is inferior to the full trajectory attack, the detector still missed many safety-critical objects yet the perturbation in point cloud space is highly imperceptible: *even for human eyes, it is quite difficult to distinguish the point cloud before and after the polynomial trajectory perturbation.*

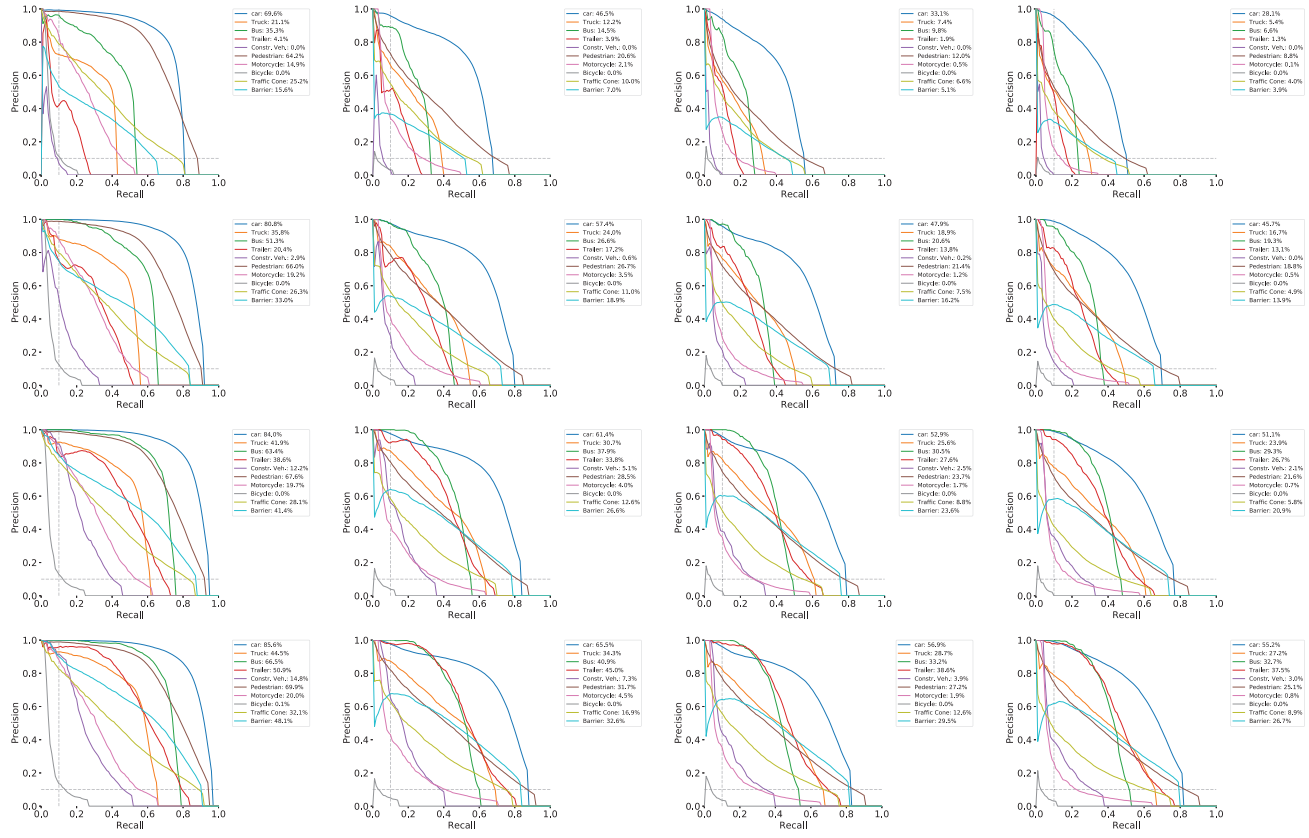
V. Summary

In this supplementary, we presented more qualitative evaluations of both white box and black box attack, to validate the effectiveness of our attack pipeline. Besides, we found that the number of PGD iteration and the step size



(a) t (b) t (c) t (d) t (e) t

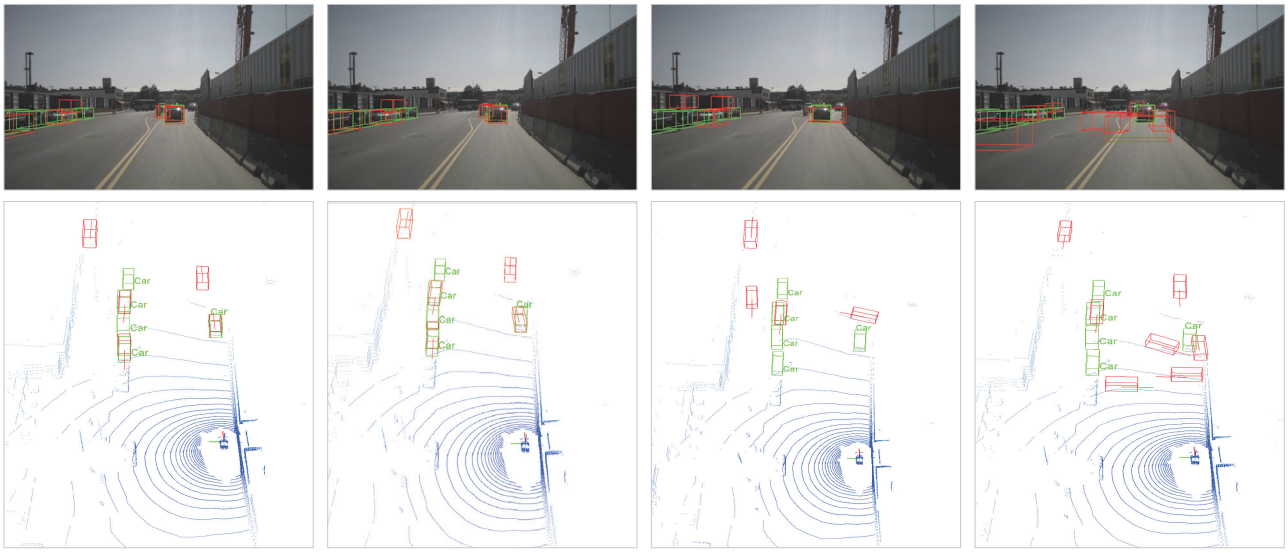
Figure VII. Per category precision-recall plot of attacking the **full trajectory** (black box) on the nuScenes validation set [2].



(a) t (b) t (c) t (d) t

Figure VIII. Per category precision-recall plot of **original detector**, attacking **translation** only, attacking **rotation** only, attacking **full trajectory** (black box) on the nuScenes validation set [2] (from left to right). From top to bottom is respectively the results from four thresholds (0.5m, 1.0m, 2.0m, 4.0m). From left to right, the curve shifts to the lower left when increasing the attack magnitude.

for each iteration have a relatively small impact on the attack quality, but the interpolation step, *i.e.*, the number of LiDAR packets, can have a relatively large influence on the attack performance, because increasing trajectory points which can be deliberately modified can facilitate the adversarial learning, resulting in a stronger adversarial perturbation. Finally, more qualitative examples of the polynomial trajectory perturbation are demonstrated to validate the imperceptibility of out attack.

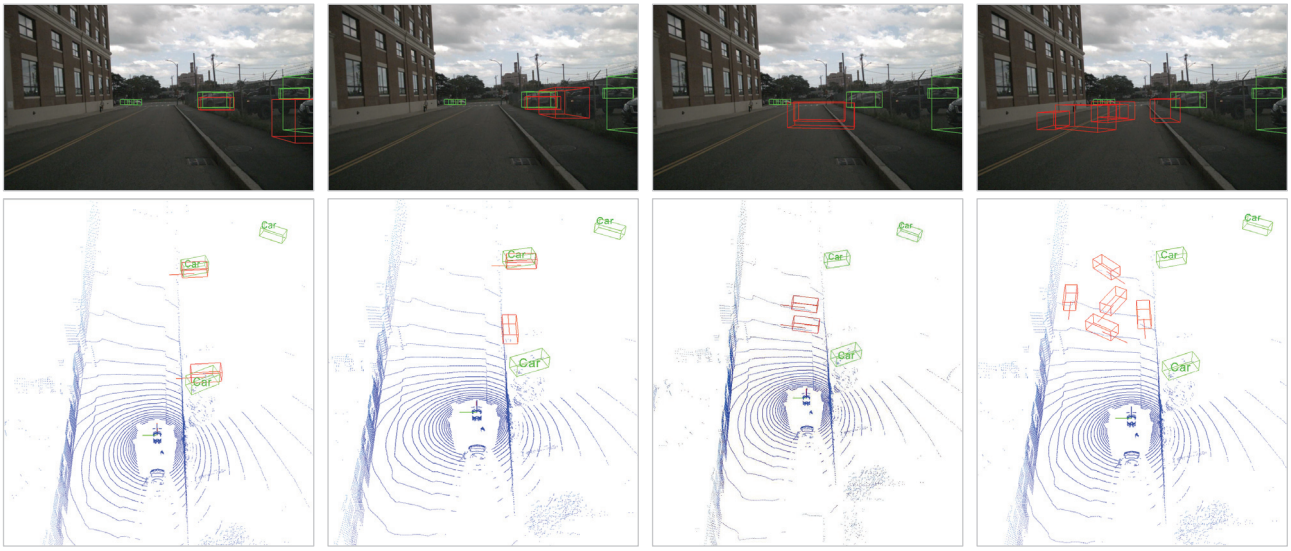


a) PointRCNN b) FLAT (Translation) c) FLAT (Rotation) d) FLAT (Full)

Figure IX. Qualitative evaluations of white box attack. False positives are increased and predictions are drifted by carefully crafting the vehicle trajectory.

Table I. 3D detection performance of white-box **translation** attack. We report the average precision of the 3D bounding box for the car category with the IoU threshold 0.7, under different levels of difficulty and ranges of depth following [28]. iter_eps and nb_iter respectively denote the step size of each iteration and the number of iteration. The best and second best attack qualities are respectively highlighted using **red** and **blue** color.

Attack Setting \ Case		iter_eps	nb_iter	Easy	Moderate	Hard	0-30m	30-50m	50-70m
None		-	-	47.44	21.56	20.91	47.44	2.16	0.17
Coordinate Attack		-	-	16.42	6.58	5.90	15.20	0.48	0.03
Random Attack		-	-	17.00	8.58	8.90	20.43	1.09	0.09
Classification	Stage-1	0.05	10	15.38	7.32	7.66	18.12	0.93	**0.03**
		0.05	20	16.85	8.49	8.69	19.87	1.29	0.05
		0.05	30	17.60	8.81	8.95	20.60	0.82	0.05
		0.1	10	13.12	6.49	6.79	15.83	**0.71**	**0.04**
		0.1	20	12.94	6.58	7.22	16.82	0.86	0.06
		0.1	30	14.77	7.14	7.69	18.57	0.74	0.05
	Stage-2	0.05	10	12.47	6.72	7.29	16.13	1.16	0.14
		0.05	20	13.89	6.82	6.96	16.11	0.95	0.17
		0.05	30	**11.87**	**5.98**	**6.40**	**14.87**	**0.67**	0.05
		0.1	10	12.58	6.40	6.77	15.25	0.95	0.07
		0.1	20	**11.72**	**5.87**	**6.06**	**13.91**	0.87	**0.04**
		0.1	30	12.54	6.52	6.69	15.14	0.96	0.09
Regression	Stage-1	0.05	10	17.45	8.43	8.42	19.53	1.12	**0.04**
		0.05	20	16.52	8.53	8.75	19.69	1.37	**0.04**
		0.05	30	17.60	8.81	9.11	21.19	0.85	**0.04**
		0.1	10	14.47	7.51	7.88	17.77	1.07	0.05
		0.1	20	17.46	8.24	8.57	19.36	1.09	**0.03**
		0.1	30	14.42	7.04	7.31	16.44	1.02	0.13
	Stage-2	0.05	10	26.07	12.76	12.51	27.19	2.16	0.17
		0.05	20	26.05	12.75	12.50	27.19	2.16	0.15
		0.05	30	26.07	12.76	12.51	27.19	2.16	0.17
		0.1	10	26.07	12.76	12.51	27.19	2.16	0.17
		0.1	20	26.09	12.78	12.53	27.15	2.17	0.17
		0.1	30	26.07	12.76	12.51	27.19	2.16	0.17

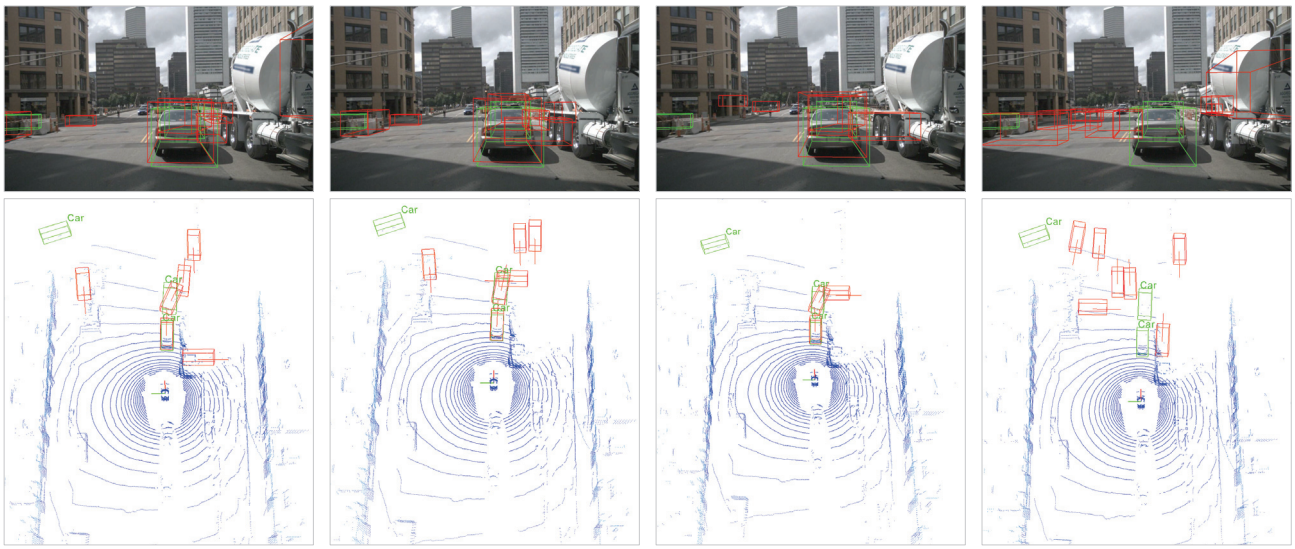


a) PointRCNN b) FLAT (Translation) c) FLAT (Rotation) d) FLAT (Full)

Figure X. Qualitative evaluations of white box attack. False positives are increased and predictions are drifted by carefully crafting the vehicle trajectory.

Table II. 3D object detection performance of white-box **rotation** attack. Other settings are similar to Table I.

Attack Setting \ Case		iter_eps	nb_iter	Easy	Moderate	Hard	0-30m	30-50m	50-70m
None		-	-	47.44	21.56	20.91	47.44	2.16	0.17
Random Attack		-	-	12.30	4.87	5.13	13.31	**0.01**	**0.00**
Classification	Stage-1	0.005	10	5.38	1.91	2.05	5.89	**0.01**	**0.00**
		0.005	20	7.16	2.94	2.81	7.75	0.02	**0.00**
		0.005	30	6.17	2.41	2.49	7.02	**0.01**	**0.00**
		0.01	10	3.93	1.70	1.61	4.79	**0.01**	**0.00**
		0.01	20	6.32	2.43	2.51	7.32	0.02	**0.00**
		0.01	30	6.83	2.53	2.27	6.87	0.02	**0.00**
	Stage-2	0.005	10	4.32	1.48	1.35	4.08	**0.00**	**0.00**
		0.005	20	3.39	1.20	1.14	3.39	**0.01**	**0.00**
		0.005	30	6.59	2.22	2.04	6.15	**0.01**	**0.00**
		0.01	10	3.18	1.20	1.17	3.44	**0.00**	**0.00**
		0.01	20	**2.35**	**0.80**	**0.61**	**2.03**	**0.01**	**0.00**
		0.01	30	**2.36**	**0.90**	**0.94**	**2.84**	**0.01**	**0.00**
Regression	Stage-1	0.005	10	8.46	3.56	3.36	9.01	0.04	**0.00**
		0.005	20	8.11	3.11	3.06	7.96	0.02	0.03
		0.005	30	8.03	3.25	3.02	8.27	**0.01**	**0.00**
		0.01	10	5.02	1.79	1.61	4.79	**0.01**	0.01
		0.01	20	5.50	1.87	1.76	5.45	0.02	**0.00**
		0.01	30	4.17	1.53	1.47	4.60	**0.01**	**0.00**
	Stage-2	0.005	10	25.99	12.71	12.49	27.07	2.16	0.17
		0.005	20	26.07	12.76	12.51	27.19	2.16	0.17
		0.005	30	26.07	12.76	12.51	27.19	2.16	0.17
		0.01	10	26.07	12.76	12.51	27.19	2.16	0.17
		0.01	20	26.30	12.89	12.59	27.35	2.17	0.17
		0.01	30	26.18	12.75	12.53	27.24	2.16	0.17



a) PointRCNN b) FLAT (Translation) c) FLAT (Rotation) d) FLAT (Full)

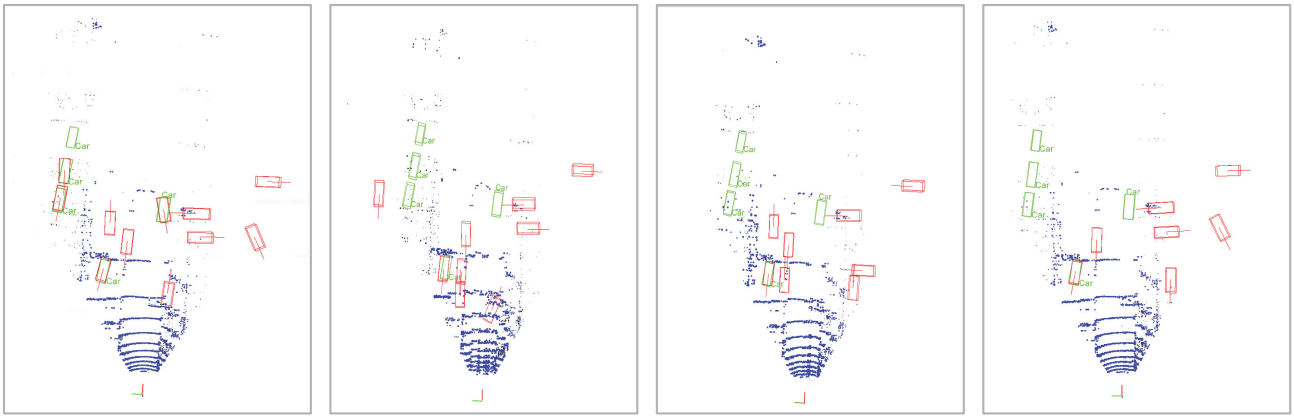
Figure XI. Qualitative evaluations of white box attack. False positives are increased and predictions are drifted by carefully crafting the vehicle trajectory.

Table III. 3D detection performance of white-box **full trajectory** attack. Other settings are similar to Table I.

Attack Setting \ Case		iter_eps	iter_eps2	nb_iter	Easy	Moderate	Hard	0-30m	30-50m	50-70m
None		-	-	-	47.44	21.56	20.91	47.44	2.16	**0.17**
Random Attack		-	-	-	5.66	2.43	2.78	7.67	0.02	**0.00**
Classification	Stage-1	0.005	0.05	10	1.11	0.25	0.23	1.19	0.02	**0.00**
		0.005	0.05	20	2.89	1.02	1.15	3.52	**0.01**	**0.00**
		0.005	0.05	30	0.65	0.21	0.25	0.88	**0.01**	**0.00**
		0.01	0.1	10	0.57	0.16	0.16	0.67	**0.01**	**0.00**
		0.01	0.1	20	1.52	0.45	0.51	1.71	**0.01**	**0.00**
		0.01	0.1	30	0.84	0.20	0.24	1.03	**0.00**	**0.00**
	Stage-2	0.005	0.05	10	0.37	0.09	0.07	0.43	**0.00**	**0.00**
		0.005	0.05	20	0.42	0.15	**0.12**	0.56	**0.00**	**0.00**
		0.005	0.05	30	**0.09**	**0.01**	**0.02**	**0.12**	**0.00**	**0.00**
		0.01	0.1	10	0.22	**0.02**	**0.02**	**0.21**	**0.00**	**0.00**
		0.01	0.1	20	**0.19**	**0.01**	**0.02**	0.26	**0.00**	**0.00**
		0.01	0.1	30	0.54	0.15	0.15	0.54	0.03	**0.00**
Regression	Stage-1	0.005	0.05	10	1.20	0.33	0.32	1.27	**0.01**	**0.00**
		0.005	0.05	20	2.44	0.90	0.88	2.72	0.02	**0.00**
		0.005	0.05	30	2.07	0.77	0.76	2.68	**0.01**	**0.00**
		0.01	0.1	10	0.90	0.36	0.35	1.18	**0.01**	**0.00**
		0.01	0.1	20	1.01	0.35	0.32	1.27	**0.01**	**0.00**
		0.01	0.1	30	0.53	0.11	0.13	0.80	0.03	**0.00**
	Stage-2	0.005	0.05	10	26.07	12.76	12.51	27.19	2.16	**0.17**
		0.005	0.05	20	26.08	12.76	12.52	27.19	2.17	**0.17**
		0.005	0.05	30	26.06	12.75	12.51	27.17	2.16	**0.17**
		0.01	0.1	10	26.07	12.76	12.51	27.19	2.16	**0.17**
		0.01	0.1	20	26.03	12.70	12.48	27.13	2.16	**0.17**
		0.01	0.1	30	26.07	12.76	12.51	27.19	2.16	**0.17**

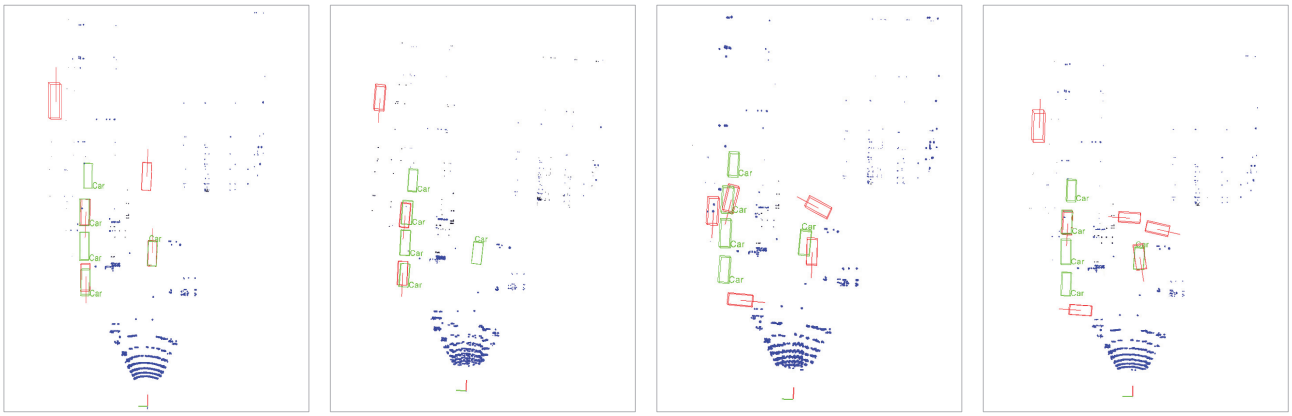
Table IV. 3D object detection performance under different interpolation steps (attacking the **classification** in **stage-2**). The number of iteration and the step size for each iteration is fixed as 20 and 0.1/0.01 (translation/rotation), respectively.

Attack Setting	Interpolation Step	Easy	Moderate	Hard	0-30m	30-50m	50-70m
FLAT (Translation)	25	16.97	7.71	7.99	18.12	0.93	0.13
	50	13.73	4.75	6.33	16.33	0.34	0.00
	100	11.72	5.87	6.06	13.91	0.87	0.04
	500	10.29	4.64	5.94	17.35	0.73	0.00
	1000	8.69	3.02	5.52	14.62	1.10	0.00
FLAT (Rotation)	25	2.53	0.89	0.75	2.18	0.00	0.00
	50	4.73	1.66	1.59	4.30	0.01	0.00
	100	2.35	0.80	0.61	2.03	0.01	0.00
	500	1.77	0.62	0.75	2.70	0.01	0.00
	1000	1.03	0.27	0.21	1.11	0.00	0.00
FLAT (Full)	25	0.36	0.13	0.14	0.40	0.00	0.00
	50	0.58	0.18	0.17	0.58	0.00	0.00
	100	0.19	0.01	0.02	0.26	0.00	0.00
	500	0.17	0.01	0.02	0.09	0.00	0.00
	1000	0.07	0.02	0.02	0.11	0.01	0.00



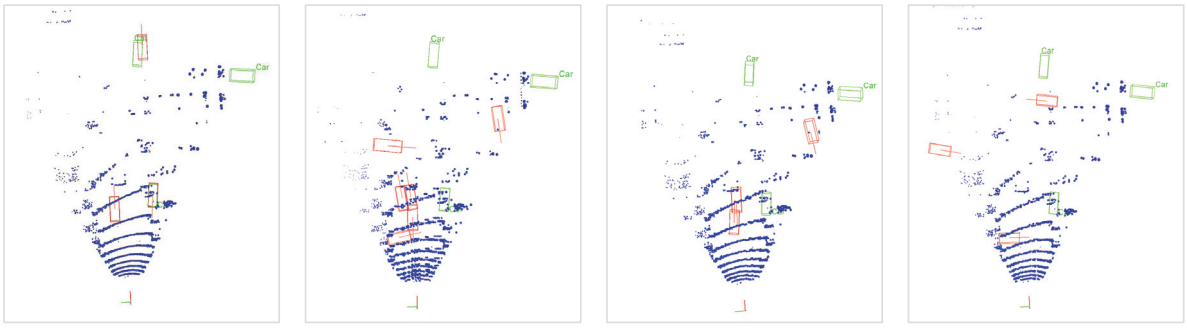
a) PointRCNN | b) FLAT (Full) w/o regularization | c) FLAT (Full) w/ regularization (λ=0.01) | d) FLAT (Full) w/ regularization (λ=1)

Figure XII. Qualitative example of FLAT attack with and without regularization. A minor perturbation in point cloud can fool the detector.



a) PointRCNN | b) FLAT (Full) w/o regularization | c) FLAT (Full) w/ regularization (λ=0.01) | d) FLAT (Full) w/ regularization (λ=1)

Figure XIII. Qualitative example of FLAT attack with and without regularization. A minor perturbation in point cloud can fool the detector.



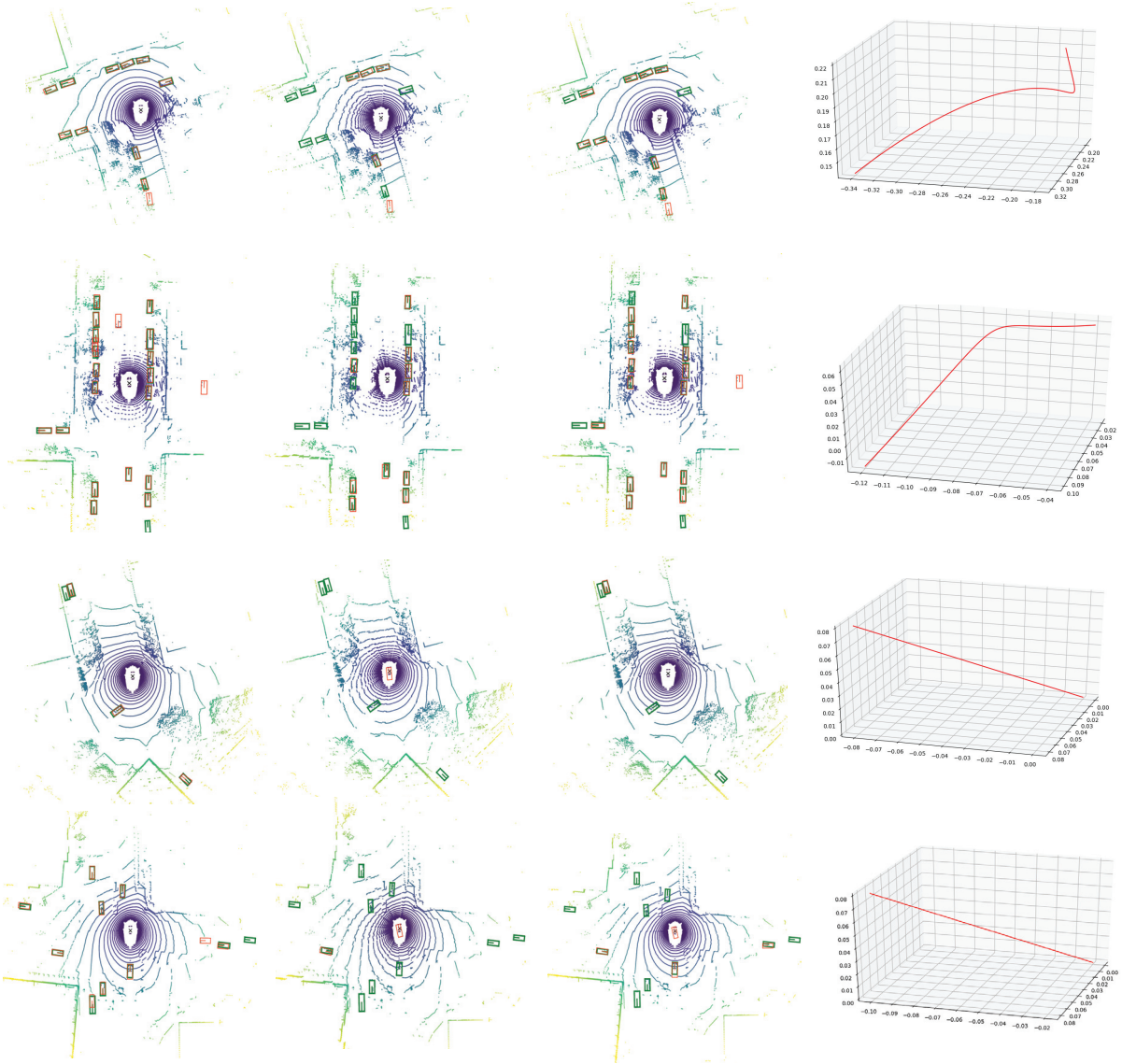
a) PointRCNN

b) FLAT (Full)
w/o regularization

c) FLAT (Full)
w/ regularization (λ=0.01)

d) FLAT (Full)
w/ regularization (λ=0.1)

Figure XIV. Qualitative example of FLAT attack with and without regularization. A minor perturbation in point cloud can fool the detector.



a) Original Predictions

b) FLAT (Full Trajectory)

c) FLAT (Polynomial)

d) Polynomial Trajectory Perturbation

Figure XV. Point cloud visualization and qualitative results of black box attack. a) Raw detections of the original detector PointPillar++ [11]. b) The output of the detector after attacking the full trajectory. c) The output of the detector after polynomial trajectory perturbation in the euclidean space. d) The polynomial translation perturbation visualized in xyz space, the units of three axes are all meters.