Supplementary Material: AdvDO: Realistic Adversarial Attacks for Trajectory Prediction

Yulong Cao[1,2], Chaowei Xiao[2], Anima Anandkumar[2,3], Danfei Xu[2], and Marco Pavone[2,4]

[1] University of Michigan, Ann Arbor
[2] NVIDIA
[3] California Institute of Technology
[4] Stanford University

A Related works

Adversarial Traffic Scenarios Generation In Strive [1], adversarial scenarios generated from the traffic model is not always realistic due to the limited training data which does not cover dangerous scenarios such as collisions. In Figure A, we demonstrated from one example generated in Strive, where the adversarial agent drives in reverse lane and violates the traffic rule, in order to collide into the AV.

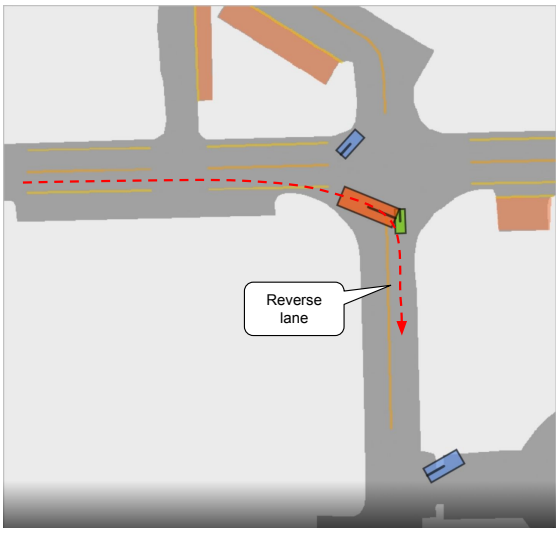


Fig. A: Adversarial agent drives in reverse lane in adversarial scenarios generatated from Strive [1].

B Method

In this section, we describe implementation and formulation details for the proposed method.

B.1 Differential dynamic model

The differential dynamic model Φ is devised for deriving dynamic parameters $\{p, v, \theta\}$ from control actions $u = \{a, \kappa\}$ and deriving control actions from trajectories $p = (p_x, p_y)$. Specifically, we use a kinematic bicycle model as the dynamic model [2]. Detailed formulation is as below:

$$
\begin{aligned}
\Phi: \quad & v^{t+1} = a^t \cdot \Delta t + v^t \\
& d\theta^t = v^t \cdot \kappa^t \\
& \theta^{t+1} = d\theta^t \cdot \Delta t + \theta^t \\
& p_x^{t+1} = v^t \cdot \cos\theta^t \cdot \Delta t + p^t \\
& p_y^{t+1} = v^t \cdot \sin\theta^t \cdot \Delta t + p^t \\
\Phi^{-1}: \quad & v^t = \|p^{t+1} - p^t\|/\Delta t \\
& \theta^t = \arctan p_x^t / p_y^t \\
& a^t = (v^{t+1} - v^t)/\Delta t \\
& \kappa^t = d\theta^t / v^t
\end{aligned}
$$

For the physical constraints for dynamically feasibility, we follow the standard values used in [3].

B.2 Reconstruction loss and adversarial loss

Here, we describe losses for reconstruction and generating adversarial trajectory in details:

$$
l_{\text{dyn}}(\theta, v, a, \kappa) = \sum_{x=\theta,v,a,\kappa} (x - x_{\text{lb}})/(x_{\text{ub}} - x_{\text{lb}}) - \text{Sigmoid}\left((x - x_{\text{lb}})/(x_{\text{ub}} - x_{\text{lb}})\right) + 0.5
$$

, where x_{ub}, x_{lb} represent the hard-coded upper bound and lower bound correspondingly for the dynamic parameter x.

$$
l_{\text{col}}(\mathbf{D}_{\text{adv}}, \mathbf{X}) = \frac{1}{n-1} \sum_{i \neq adv}^{n-1} \frac{1}{\|\mathbf{D}_{\text{adv}} - \mathbf{X}_i\| + 1}
$$

, where n is the number of agent in the current prediction time frame.

$$
l_{\text{bh}}(\mathbf{D}_{\text{adv}}, \mathbf{D^*}_{\text{orig}}, \epsilon) = \|\mathbf{D}_{\text{adv}} - \mathbf{D^*}_{\text{orig}}\|/\epsilon - \text{Sigmoid}\left(\|\mathbf{D}_{\text{adv}} - \mathbf{D^*}_{\text{orig}}\|/\epsilon\right) + 0.5
$$

, where ϵ is the tolerance for position deviation, which we empirically set to half lane width (1 meter).

$$l_{\text{obj}} = \frac{1}{T} \sum_{t=1...T} \parallel \mathbf{Y^t} - \hat{\mathbf{Y}}^{\mathbf{t}} \parallel_2$$

, where $\hat{\mathbf{Y}}^t$ is the predicted future trajectory at time t given the adversarial trajectory and $\mathbf{Y}^t$ is the corresponding ground truth. This loss aims to mislead the prediction by maximizing the difference between the predicted future trajectory and ground truth.

C Experiments

In this section, we describe implementation and formulation details for the experiments.

C.1 Attack fidelity analysis

In this analysis, we aim to demonstrate the generated adversarial trajectory is realistic from both perspectives of: (1) dynamically feasibility and (2) similar behavior as the original history trajectory. For the first perspective, we demonstrate the results quantitatively with the **Violation Rates** (VR) metric described below. For the second perspective, since it is a common challenge to measure the behavior change quantitatively, we propose to approximate the degree of behavior change with the **Aggregated sensitivity** metric described below. We also visually examine generated adversarial trajectories in Figure B.

Violation rates. Since the violation rates metric is only suitable for the *search* method and on the curvature κ parameter, we represent the VR as:

$$VR = \frac{\#\text{total adv trajectories}}{\#\text{adv trajectories violating curvature constraints}}$$

.

Aggregated sensitivity. To approximate the behavior change quantitatively, we leverage the sensitivity concept proposed by Ivanovic et al. [4]. Sensitivity $\mathbf{PI}(\mathbf{Y}_i, \mathbf{Y}_{\text{ego}})$ of an agent's trajectory to the ego agent represents how much the agent's trajectory $\mathbf{Y}_i$ will affect the ego planning $\mathbf{Y}_{\text{ego}}$. Therefore, we can present how much the adversarial trajectory $\mathbf{X}_{\text{adv}}$ will affect other agents' planning $\mathbf{X}_i$ as the aggregated sensitivity of the adversarial agent's trajectory to all the other agents in the scene. With a normalization over agents nearby, we attain the aggregated sensitivity:

$$\Sigma\text{Sensitivity}(\mathbf{X}_{\text{adv}}, \mathbf{X}) = \frac{1}{m} \sum_{i, \|\mathbf{X}_{\text{adv}} - \mathbf{X}_i\| < \rho}^{m} \mathbf{PI}(\mathbf{X}_{\text{adv}}, \mathbf{X}_i)$$

, where m represents the total number of agents nearby filtered by the distance threshold ρ, which is empirically set to 5 meters. Therefore, we attain the metric for measuring behavior change as:

$$\Delta\text{Sensitivity} = \Sigma\text{Sensitivity}(\mathbf{X}_{\text{adv}}, \mathbf{X}) - \Sigma\text{Sensitivity}(\mathbf{X}_{\text{orig}}, \mathbf{X})$$

Other metrics. To measure the behavior change quantitatively, we also include evaluation results with other metrics proposed by Jekel et al. for comparing the similarity between trajectories [5], including Dynamic Time Warping (DTW), Fréchet Distance (FD), Partial Curve Mapping (PCM), Area and Curve Length (CL). In Table A we demonstrate that the proposed methods have lowest error for all similarity metrics. The results are also consistent with the result on Δsensitivity metric.

Table A: Similarity between original history trajectory and adversarial trajectory generated from *search*, *Opt-init* and *Opt-end*.

Attack method	DTW↓	FD↓	PCM↓	Area↓	CL↓
search	0.3558	0.2490	0.0676	0.8892	0.0003
Opt-init	0.2303	0.1429	0.0209	0.5928	0.0002
Opt-end	0.1891	0.0564	0.0210	0.3045	0.0001

Table B: Augmentation on AgentFormer.

	ADE	FDE
Benign	1.83	3.81
+ *aug*	1.69	3.57

Visualization. We randomly sample examples from 150 scenes in nuScene validation data, where the adversarial trajectory generated from *search* that have a curvature violation or a large ΔSensitivity value. In Figure B, we show that the adversarial trajectory generated from *search* have either behavior change or unrealistic steering rates. We also notice that, the *Opt-end* can also generate adversarial trajectory that has large turning rates but dynamically feasible. Even though the predicted results are worse under *search* attack when the curvature constraint if not bounded, *Opt-end* achieves higher prediction errors in average scenarios. To further show that the generated trajectories obey traffic rules, we conduct a study where adversarial trajectories are illustrated with map information (e.g. lane segments, road, crosswalk etc.). We select five human subjects with driver license and show our generated trajectories to them. Out of the 50 trajectories evaluated, only 2.2($\pm$1.3) are considered rule-violating. We conclude that the adversarial trajectory generated by our methods are more realistic in both perspectives of dynamical feasibility and behavior changing.

AdvDO as Augmentation. Noticed that AdvDO also provides a framework for generating realistic trajectories. We replace the adversarial objective losses with other objectives (e.g. left/right/forward/backward deviations) and generate additional data. In Table B, we demonstrate that the augmented data improves the clean performance by 9% on ADE. This further validates that the high fidelity of the generated trajectories with the proposed method.

C.2 Case studies with planners

Planner. In this work, to demonstrate the explicit consequences of the adversarial trajectory, we implement two planners (including path planning and motion planning). The first one is a rule-based planner as implemented by Rempe et al. [1]. However, we notice that this planner is enforcing path planning along the center of lane lines which leads to insufficient path sampling through the simulations. Therefore, though the planner naturally avoids driving off road, it is also lack of flexibility to dodge incoming traffic. To better represent planners equipped on AV, we implement a simple yet effective planner that uses conformal lattice [6] for sampling paths and model predictive control (MPC) [7] for motion planning. We call this planner MPC-based planner.
Planning strategy. In this work, we consider both an open-loop and a closed-loop planning strategy. Though for the closed-loop planning we have to replay the ground truth trajectories of other agents, we do notice reduced collisions and driving off road consequences and consider the closed-loop planning fashion meaningful.

C.3 Transferability Analysis

In this section, we aim to analyze the transferability of adversarial trajectories generated on a source model to a unseen target model. We measure the transferability by devising the **transfer rate** metric. High transfer rates indicate that the feasibility of transfer attack, which is a more realistic black-box attack, in the real-world scenario. Transfer rate is defined as the success degree of adversarial trajectories on target model over the success degree of them on source model. The success degree is measured by the average percentage of increased error (on metrics ADE/FDE/MR/ORR) with transfer attack on the target models over the increased error with white-box attack on the source models.

C.4 Ablation Study

We explore the attack results in different traffic scenarios with different speeds curvatures. We calculate the aggregated speed and curvature for each agent in the entire scene to represent the speed and curvature for that scene. Similarly, we calculate the aggregated *Miss Rates* to evaluate performance.
Attack effectiveness with different speeds As shown in Figure Ca, the higher speed traffic show higher *Miss Rates*. It is reasonable since position deviations are larger in high speed traffics. We also notice that the attack results are consistent to results in Table 1&2 in the main paper, which means different attack methods are not restricted due to the speed constraints.
Attack effectiveness with different curvatures In Figure Cb, we notice that adversarial trajectories are more effective in small curvature traffics. This is reasonable since small curvature traffics allow more flexible adversarial trajectory generations. We find that *Opt-end* performs better than *Opt-init* in small curvature traffic. This could be due to low curvature traffic being less sensitive to current positions.

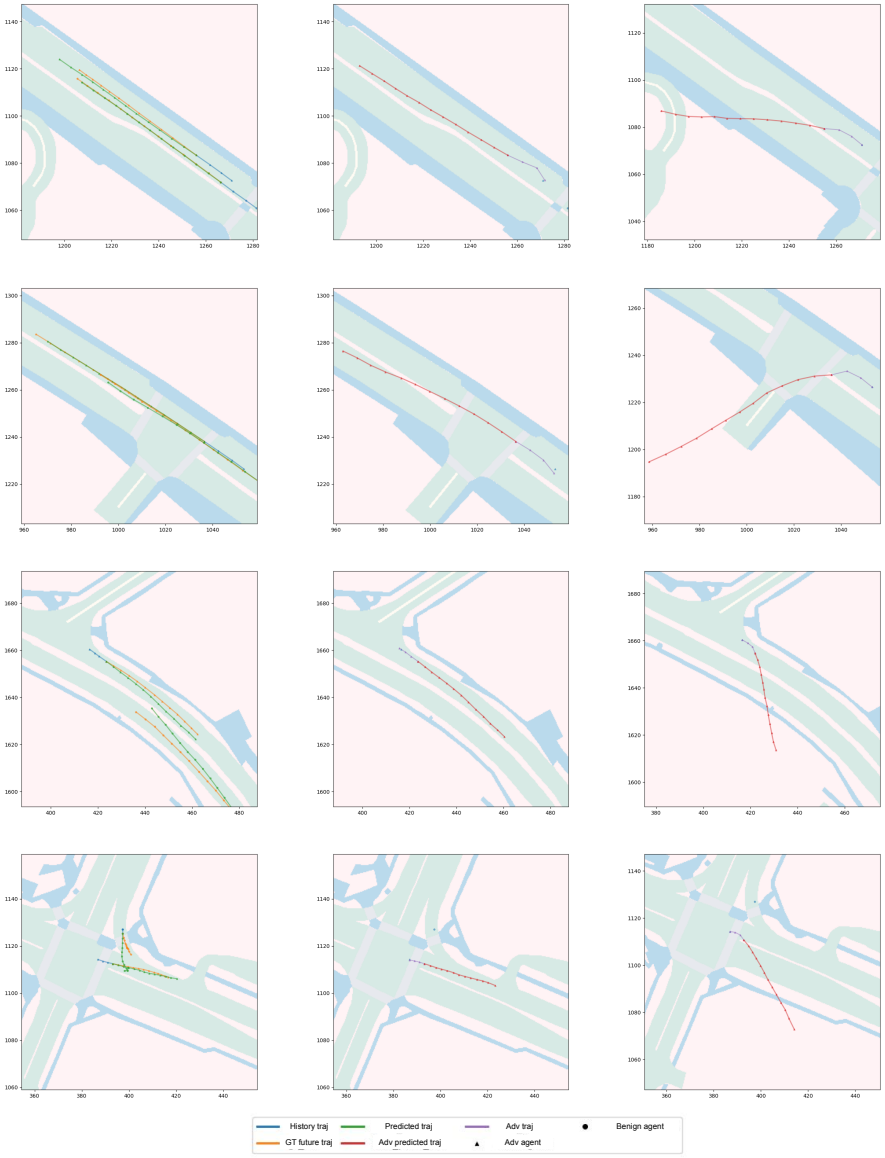
C.5 Mitigation

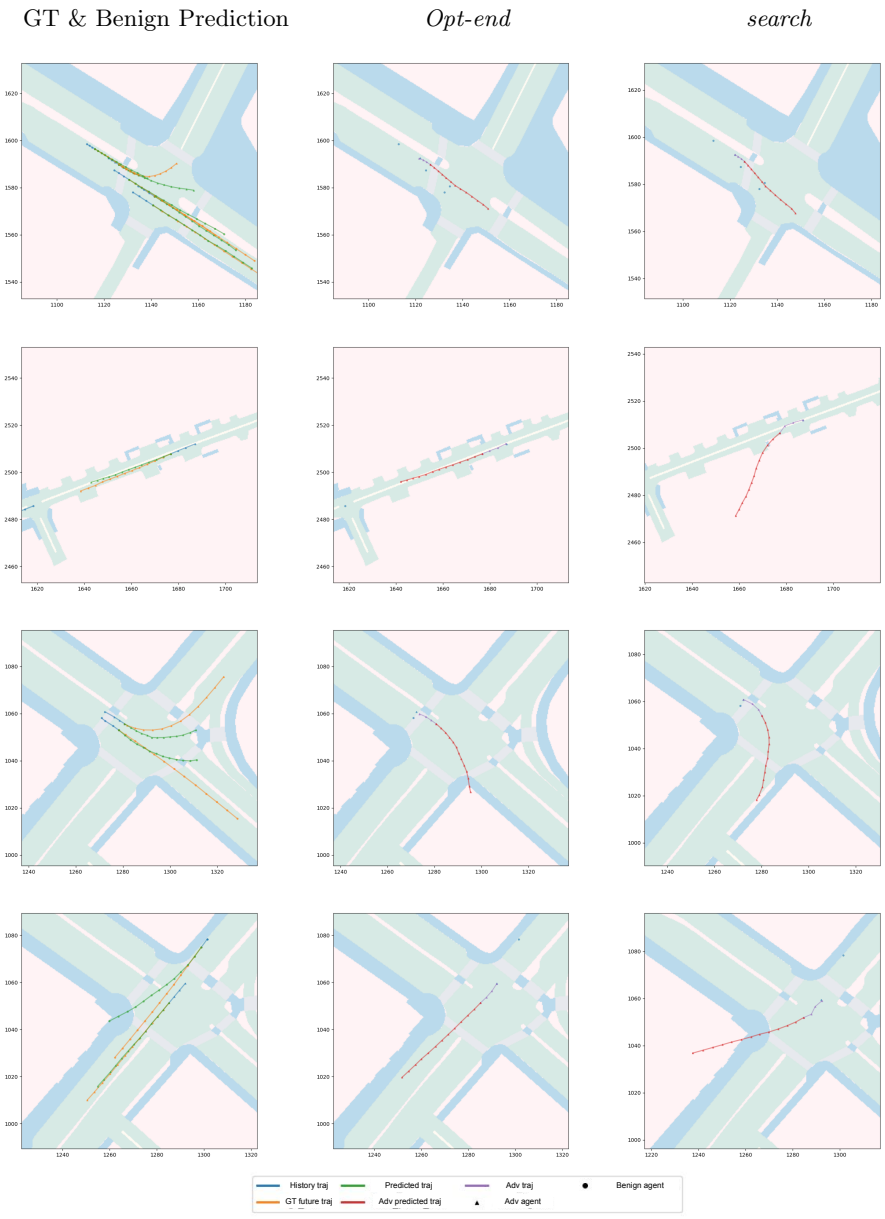
We present a preliminary mitigation methods against adversarial trajectory via adversarial training. We notice that naive adversarial training results in noticeable degradation in benign performance for both adversarial trained models using *search* and *Opt-init*. In Table C, we demonstrate that the performance degradation are much smaller and even better for the adversarial trained model with proposed method *Opt-init*.

Table C: Adversarial training results. The number in brackets represent the difference between the benign model and adversarial trained model.

Model	Attack	ADE ↓	FDE ↓	MR ↓	ORR ↓
Benign	Benign	1.83	3.81	28.2%	4.7%
	search	2.34	4.78	34.3%	6.6%
	Opt-init	3.39	5.75	44.0%	10.4%
Rob-search	Benign	2.69(+0.86)	5.82(+2.01)	37.8% (+9.6%)	10.2%(+5.5%)
	search	2.72(+0.38)	5.76(+0.98)	40.7%(+6.3%)	12.3%(+5.8%)
	Opt-init	2.81(-0.58)	5.92(+0.17)	42.2%(-1.8%)	13.8%(+3.4%)
Rob-ours	Benign	2.38(+0.55)	5.03(+1.23)	35.1%(+6.9%)	8.1%(+3.4%)
	search	2.42(+0.08)	5.25(+0.47)	36.8%(+2.5%)	9.2%(+2.6%)
	Opt-init	2.54(-0.85)	5.21(-0.54)	36.4%(-7.6%)	8.9%(-1.5%)

GT & Benign Prediction *Opt-end* *search*




GT & Benign Prediction
Opt-end
search
History traj
GT future traj
Predicted traj
Adv predicted traj
Adv traj
Adv agent
Benign agent

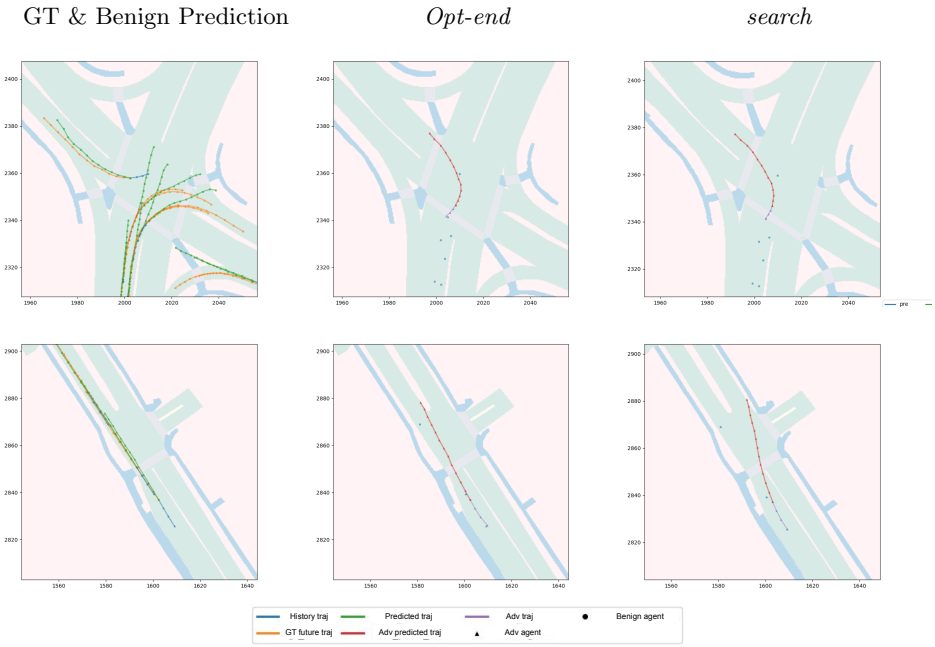
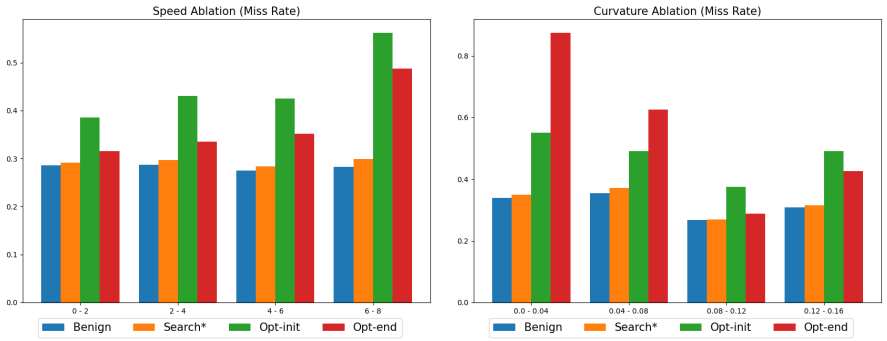


Fig. B: Visualization examples of generated adversarial trajectories from *Opt-end* and *search*. We only show the adversarial agent's trajectory in the attack scenario for clearer visualization.



(a) Speed ablation (b) Curvature ablation

Fig. C: Ablation studies for different traffic scenes.

References

1. Davis Rempe, Jonah Philion, Leonidas J. Guibas, Sanja Fidler, and Or Litany. Generating useful accident-prone driving scenarios via a learned traffic prior. In *arXiv:2112.05077*, 2021.
2. OA Condrea, A Chiru, RL Chiriac, and S Vlase. Mathematical model for studying cyclist kinematics in vehicle-bicycle frontal collisions. *IOP Conference Series: Materials Science and Engineering*, 252:012003, oct 2017.
3. 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. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
4. Boris Ivanovic and Marco Pavone. Injecting planning-awareness into prediction and detection evaluation. *CoRR*, abs/2110.03270, 2021.
5. Charles F Jekel, Gerhard Venter, Martin P Venter, Nielen Stander, and Raphael T Haftka. Similarity measures for identifying material parameters from hysteresis loops using inverse analysis. *International Journal of Material Forming*, may 2019.
6. Matthew McNaughton, Chris Urmson, John M Dolan, and Jin-Woo Lee. Motion planning for autonomous driving with a conformal spatiotemporal lattice. In *2011 IEEE International Conference on Robotics and Automation*, pages 4889–4895. IEEE, 2011.
7. Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer science & business media, 2013.