Robustness of Trajectory Prediction Models Under Map-Based Attacks

Zhihao Zheng
Lehigh University
zhzc21@lehigh.edu

Xiaowen Ying
Lehigh University
xiy517@lehigh.edu

Zhen Yao
Lehigh University
zhy321@lehigh.edu

Mooi Choo Chuah
Lehigh University
chuah@cse.lehigh.edu

Abstract

Trajectory Prediction (TP) is a critical component in the control system of an Autonomous Vehicle (AV). It predicts future motion of traffic agents based on observations of their past trajectories. Existing works have studied the vulnerability of TP models when the perception systems are under attacks and proposed corresponding mitigation schemes. Recent TP designs have incorporated context map information for performance enhancements. Such designs are subjected to a new type of attacks where an attacker can interfere with these TP models by attacking the context maps. In this paper, we study the robustness of TP models under our newly proposed map-based adversarial attacks. We show that such attacks can compromise state-of-the-art TP models that use either image-based or node-based map representation while keeping the adversarial examples imperceptible. We also demonstrate that our attacks can still be launched under the black-box settings without any knowledge of the TP models running underneath. Our experiments on the NuScene dataset show that the proposed map-based attacks can increase the trajectory prediction errors by 29-110%. Finally, we demonstrate that two defense mechanisms are effective in defending against such map-based attacks.

1. Introduction

Autonomous vehicles (AVs) are becoming more acceptable to general populations and can transform future transportation systems e.g. having autonomous trucks to deliver goods helps alleviate the on-going acute shortage of truck driver problem. A typical AV system consists of three core modules: a Perception module that interprets the surrounding environment such as traffic agents and road condition, a Prediction module that predicts the future of the environments based on history observations, and a Planning module that aggregate all the information to decide how to control and navigate the AV. Within the Prediction module, Trajectory Prediction (TP) is one of the most important tasks that aims to forecast the motion of surrounding traffic agents.

Many recent studies have explored deep neural network based trajectory prediction models. Researchers evaluate these models based on well known benchmarks collected from real world such as Kitti[9], Apolloscape[13], NuScenes [1]. The metrics that are often used by these researchers include ADE (average displacement errors between ground truth and predicted trajectories in a predicted time window) and FDE (final displacement errors at the end of the predicted time window). Recent trajectory prediction models [24, 28, 10, 11, 23] that perform well typically use CVAE (to generate K likely trajectories) and semantic map to provide better context for the trajectory predictor.

In recent years, researchers conduct studies to explore the robustness of DNN models used in autonomous vehicles. Researchers have demonstrated that they can fool the object detector and lane detection subsystem in the perception module [8, 2, 25]. They have also shown that both the monocular and LIDAR-based depth estimation submodules can be attacked as well. Most of these attacks have been demonstrated only on certain submodules of the perception systems. As far as we know, there is only one recent paper [29] that studies the robustness of the trajectory prediction module. In [29], the authors propose white/black box adversarial attacks on trajectory prediction by adding minor perturbations on normal trajectories to maximize the prediction error. Their attacks are designed to make adversarial trajectories look natural by obeying physical rules. They also define optimization objectives that allow predicted trajectories to deviate laterally or longitudinally in order to create potential danger in AV driving behaviors. However, their attack approach may not be attractive for the attackers as implementing such attack may impose danger to the attackers themselves.

In this paper, we explore a different attack approach on trajectory prediction models. Our attack approach is geared towards those models that utilize context maps e.g. [24, 28, 6, 10, 11, 23] by inserting minor perturbations to the context maps these models use. There are two main categories of trajectory prediction models utilizing context maps depending on how they encode the map information, namely (i) image-based encoding for map, (ii) node-based encoding for map. We select 2 representative schemes for

each category and describe how we can launch both white-box and black-box attacks on these models.

Subsequently, we evaluate the proposed attacks using nuScenes dataset. Our results show that our semantic map attack approach significantly degrades the prediction performance of all 4 map-based trajectory prediction models. We also conduct sensitivity analysis of different factors that affect the attack results and provide some visualizations to illustrate the attack impacts. Finally, we show the effectiveness of two defense mechanisms against such attacks.

2. Related Work

Trajectory Prediction (TP) The trajectory prediction submodule in the control system of an AV predicts future spatial coordinates of nearby traffic agents such as pedestrians or vehicles. Typically trajectory prediction models are based on deep neural networks which take location coordinates of traffic agents for the past few seconds as input and may also incorporate additional information e.g. vehicles' heading, interactions among different traffic agents or semantic maps to improve the prediction performance.

Various approaches utilizing merely monocular images for future agents motion prediction have been proposed, Trajectron++ [24] authors designed a graph-structured generative (CVAE-based) neural architecture that forecasts the trajectories of diverse agents while incorporating agent dynamics. Another approach, AgentFormer [28], proposed a new Transformer that simultaneously models the temporal and social dimensions of multi-agent trajectories and a novel agent-aware attention mechanism for stochastic multi-agent trajectory prediction. GOHOME [10] leverages graph representations of the High Definition(HD) Map and sparse projections to generate a heatmap output representing future position probabiliity distribution of a given agent in a traffic scene. The authors later design THOMAS [11] which encodes past trajectories of all agents present in the scene and the HD-Map lanelet graph and predicts for each agent a sparse heatpmap representing the future probability distribution at a fixed time step in future. They showed that the performance of THOMAS is better than GOHOME.

Adversarial Attacks against Perception System in AVs Recent papers have demonstrated different types of attacks against the perception system in AVs. Researchers have demonstrated that adding perturbations to both RGB images or point clouds can cause problems to the the AV's perception systems e.g. such perturbations may cause the object detector module not being able to detect traffic agents or they may launch attacks to the depth estimator submodule to make the perception system think that the traffic agents are either further or closer. In [31], the authors presented systematic solutions to create robust adversarial example (AE)s against real world object detectors. In [22], the authors propose split-second phantom attacks to trick two commercial advanced driver assistant systems to treat a depthless object that appears for a few milliseconds as a real object and hence force such systems to stop in the middle of the road or issue false notifications.

Instead of merely attacking object detector in AV's perception system, one can add perturbations to mess up the depth estimation submodule. Authors in [27] have shown that the depth estimation accuracy in existing monocular depth estimation models degrade when subjected to typical adversarial attack methods such as IFGSM (or equivalently PGD). Apart from launching attacks on camera images to fool the object detector module in the perception system of an AV, researchers also explore the impact of attacks on the lane detection submodule within the AV perception system. In [14], the authors determines optimal perturbations they can add on a camera image which will result in errors in the lane detection submodule and then maps such perturbations to road markings in the physical world. Their extensive experimental results demonstrate that their attacks can fool the lane detection submodule of a Tesla Model S.

Fewer researchers investigate how to attack perception systems by meddling with trajectory prediction related information. In [18], the authors demonstrate that adversarial spoofing of AV's trajectory with small perturbations can make safety-critical objects undetectable or being detected with incorrect positions. In their approach, they represent point cloud as a function of trajectory and attack the trajectory instead of 3D points. In [29], the authors explore the robustness of trajectory prediction models in the presence of attacks that perturb normal vehicle trajectories. Their experiments reveal that three trajectory prediction models have significant prediction errors under such attacks.

3. Preliminary

In this section, we give a brief description of how current state-of-the-art TP models incorporate the context map into their models. These preliminary serves as the foundation for designing our map-based attacks.

3.1. Relevant Works

The **Context Map** is important for Autonomous Driving for it provides rich semantic information (e.g. drivable area, stop line, and crosswalks) that allows AVs to localize themselves and accurately navigate on their lanes. It has been recently introduced to the TP task [30, 3, 4, 12, 21, 10, 11, 6] to help models make better predictions. For example, a surrounding car is more likely to stay in its lane in the next few seconds than driving onto the sidewalk. Among all state-of-the-art TP solutions, there are two major ways of representing a context map: (1) Image Representation and (2) Node Representation. The former represents the context map as a multi-channel image where each channel corresponds to one type of semantic information, and the latter

represents each element in the context map (e.g. centerlines, stop signs) as a node. In general, image representation has higher dimensions that contain more information while node representations are more compact and efficient.

Methods that use Image Representation. Early works in trajectory prediction task [17, 30, 3, 4, 12, 21] focused on the rendering the context information as 2D top-down map image and uses Convolutional Neural Networks (CNNs) to extract features from the map. In the map image, different types of elements are rendered into different layers with binary values. Among all the TP methods that use Image Representation, we select two representative works — Trajectron++ [24] and Agentformer [28] for our experiments due to the availability of their source codes.

Although having completely different high-level designs, the map encoders in Trajectron++ [24] and Agentformer [28] are indeed very similar — based on the rendered context map, they first crop an agent-centric local map for each agent and rotate it based on the agent's heading. This rotated local context map is then fed into a CNN-based map encoder to extract a map embedding. Finally, this map embedding is concatenated with other features before feeding into the trajectory decoder to predict its future trajectories.

Methods that use Node Representation. Recently works have explored node representation for context map [11] [10] [26] [19] due to its compactness and efficiency. In this paper, we choose two representative works — LaPred [16] and PGP [6] whose source codes are publicly available.

LaPred [16] considers the context map as the set of all lane instances. Each lane instance is represented by a sequence of coordinates that are equally spaced and have similar lengths. They first employ a combination of 1D-CNN and LSTM layers to encode these coordinate sequences and produce the node features for each lane. Then, each lane feature (equivalently context information) will be merged with the features of its neighboring traffic agents and passed to the next modules.

PGP [6] represents the context map as lane graphs where each node corresponds to a lane centerline. Similar to LaPred, they also divide longer lane centerlines into smaller segments with fixed lengths except that PGP additionally includes the pose (yaw angle) of the segment as well as two binary features indicating whether the segment lies on a stop line or crosswalk. In other words, the node features in PGP capture both the geometry and the traffic control elements relevant to the centerlines. In addition, they proposed to fuse the features of the neighboring agents into the features of the lane nodes using the recently popular Attention techniques in their Node-agent Attention module.

3.2. Threat Model

In this paper, we focus on exploring the robustness of state-of-art trajectory prediction models by implementing adversarial attacks on the incorporated context maps. Since the context map provides important semantic information for AVs, it must be accurate and updated in time. Thus, the AV needs to download the context map from the data server frequently to keep the map updated irrespective of whether it loads context map online or offline.

We assume that the attacker has access to the data server and is capable of modifying the context map which will be downloaded by the victim AV. The attacker aims to add imperceptible perturbations on the correct map so that the trajectory prediction model makes wrong future predictions with the poisoned map, which may cause more dangerous reactions by the victim AV.

In real world attacks, the attacker needs to access all parameters of the prediction model from the victim AV for white box attacks or only APIs of the prediction model for black box attacks. In addition, it is more practicable for the attacker to choose a small area within the map to add adversarial perturbation considering the high computational cost and training time: the larger area for perturbing, the longer training time needed to generate effective adversarial perturbation.

4. Proposed Method

In this section, we introduce the formulation of the adversarial attacks on context maps in TP models.

4.1. Problem Formulation

State-of-art trajectory prediction models make stochastic predictions for each agent (i.e., vehicles or pedestrians) at each time frame based on the observation of all nearby agents and the context map in surrounding area. Let s_i^t be the state of an agent i at time frame t and x_i^t be the local context map surround agent i at time t. A trajectory of agent i from time t_1 to t_2 will be $S_i^{t_1,t_2} = \{s_i^{t_1}, ..., s_i^{t_2}\}$. Let N_h and N_f be the number of frames in history and future trajectories. Then at time t, the history trajectory of agent i will be $H_i^t = S_i^{t-N_h+1,t} = \{s_i^{t-N_h+1}, ..., s_i^t\}$; the future ground truth trajectory of agent i will be $F_i^t = \{s_i^{t+1}, ..., s_i^{t+N_f}\}$. Let f represent the trajectory predictor and $P_i^t = \{p_i^{t,(1)}, ..., p_i^{t,(k)}\}$ be a set of k predicted states of the agent i at time t. Thus, we denote the stochastic predicted trajectories of agent i at time i as $f(H_i^t, x_i^t) = \{P_i^{t+1}, ..., P_i^{t+N_f}\}$.

In this work, we introduce non-targeted attacks on trajectory prediction models by adding adversarial perturbation to local context maps which leads to victim models generating wrong trajectory predictions. We denote δ as adversarial perturbation and ${}^{adv}x_i^t = x_i^t + \delta$ as the adversarial context map of agent i at time t. To make the adversarial perturbation imperceptible, we restrict the perturbation with a given constraint ϵ either using Eq. 3 or Eq. 4.

To maximize the impact of attacks, we utilize two commonly used evaluation metrics in trajectory prediction tasks: (1) the minimum Average Displacement Error (**ADE**) over the top k predictions (**ADE**$_k$): the minimum average of mean L2 distance between all predicted and ground truth trajectory points. (2) the minimum Final Displacement Error (**FDE**) over the top k predictions (**ADE**$_k$): the minimum of the mean L2 distance between the predicted and ground truth trajectory points at the last time frame. Combining evaluation metrics **ADE**$_k$ and **FDE**$_k$, we denote the optimization loss as $L(f(H_i^t, x_i^t))$:

$$L(f(H_i^t, x_i^t)) = E_{ade}(F_i^t, P_i^t) + E_{fde}(F_i^t, P_i^t) \quad (1)$$

where E_{ade} and E_{fde} are the **ADE**$_k$ and **FDE**$_k$ evaluation metrics with a given k.

The goal of our attacks is to maximize the error between the predicted and ground truth trajectories by generating adversarial context maps surrounding target agents. Thus, the objective is formulated as:

$$max \quad L(f(H_i^t, {}^{adv}x_i^t)) \quad s.t. \quad C(x_i^t, {}^{adv}x_i^t) < \epsilon \quad (2)$$

4.2. Adversarial Perturbation

As mentioned in Section 3.1 , modern trajectory prediction models extract spatial information for target agents from surrounding context maps, which are mainly stored as image-based and node-based maps. We generate adversarial perturbation for both image-based and node-based context maps to launch adversarial attacks on state-of-art trajectory prediction models. To ensure changes in adversarial maps are imperceptible, we need to constrain any generated adversarial perturbation.

4.2.1 Image-based Map

Many trajectory prediction models with CNN-based encoders use image-based context maps as the extra spatial information [24, 28]. Considering the storage space of large maps and the transferability among different formats, image-based context maps are more likely to be stored as binary images [1]. Thus, we design both continuous and binary perturbation for binary maps.(i.e., $\{0,1\}$)

Continuous Perturbation. Continuous perturbation is minor continuous imperceptible perturbation that we added to the original map images. We denote the imperceptible constraint for continues perturbation as follows:

$$C(x_i^t, {}^{adv}x_i^t) = \|x_i^t - {}^{adv}x_i^t\|_\infty < \epsilon \quad (3)$$

Binary Perturbation. Minor continuous perturbations can be removed easily via filtering. Thus, to make our attacks more robust on binary map images, we also design binary perturbation that changes the binary values of a few pixels in the map. To satisfy the invisible requirement, we restrict the number of modified pixels within a given local map size:

$$C(x_i^t, {}^{adv}x_i^t) = \frac{\|x_i^t - {}^{adv}x_i^t\|_1}{v} < \epsilon \quad (4)$$

where v is the difference between binary values.

4.2.2 Node-based Map.

In recent works, some trajectory prediction models concatenate lane-based or node-based map features with agent states so as to improve the precision of predictions [6, 16]. Such node-based context maps save map information as nodes'attributes, which include spatial coordinates, lane rotations, traffic signs and so on.

We generate adversarial maps by adding continuous perturbation to the original map nodes to ensure their imperceptibility. The perturbation only changes the spatial coordinates of the map nodes (x-y location). The constraint for node-based perturbation is the same as in Eq. 3.

4.3. Attack Methods

In order to investigate the impact of adversarial attacks on trajectory prediction models with context maps, we design both white box and black box attacks following the definitions in Section 4.1 and 4.2.

4.3.1 White Box Attacks

We design our white box adversarial attacks based on Gradient-based methods [5].

Continuous Perturbation. We generate continuous adversarial perturbation on both **image-based** and **node-based** maps with Projected Gradient Descent (PGD) [20]. For agent i at time frame t, we initialize an adversarial context map ${}_0^{adv}x_i^t = x_i^t$. Let ${}_m^{adv}x_i^t$ be the m^{th} adversarial map, we update it iteratively:

$${}_{m+1}^{adv}x_i^t = \prod_x^\epsilon({}_m^{adv}x_i^t + \alpha \cdot sign(\nabla_x L(f(H_i^t, {}_m^{adv}x_i^t)))) \quad (5)$$

where α is the step size and $sign(\cdot)$ denotes the sign function. In common, we set the step size $\alpha = \frac{\epsilon}{10}$. $\prod_x^\epsilon(\cdot)$ constrains the adversarial example within the ϵ-ball of the original context map x_i^t.

Binary Perturbation. We design another white box attack against TP models with **image-based** maps by generating binary perturbation. Similar to the continuous perturbation generation, we iteratively update the binary adversarial map based on iterative gradient-descent method.

Let ${}_0^{adv}x_i^t = x_i^t$ be the initialization adversarial map for agent i at time frame t. At the m^{th} iteration, we choose the top q gradient of the adversarial map ${}_m^{adv}x_i^t$ to generate binary adversarial perturbation δ_{m+1} by:

$$\delta_{m+1} = v \cdot sign(TOP_q(\|\nabla_x L(f(H_i^t, {}_m^{adv}x_i^t)\|_\infty)) \quad (6)$$

Parameters	Definition
x_loc	x location of a patch center in the map image
y_loc	y location of a patch center in the map image
h	length of a patch
w	width of a patch
r	rotation angle of a patch

Table 1. Parameters for perturbation patches on Binary Map

Map encoding	Attacker	Perturbation	Iterations	ϵ	Particles
Image	PGD	Continuous	100	0.03	N/A
	PGD	Binary	100	50	N/A
	PSO	Binary	30	50	50
Node	PGD	Continuous	100	1.0	N/A
	PSO	Continuous	100	1.0	100

Table 2. Attack settings

where $TOP_q(\cdot)$ remains the top q pixel value and clear other pixel value. Similar to the step size α, we also set $q = \frac{\epsilon}{10}$. Then, we update the adversarial map as follows:

$$ {}^{adv}_{m+1}x_i^t = {}^{adv}_{m}x_i^t + \delta_{m+1} \tag{7} $$

4.3.2 Black Box Attacks

As trajectory prediction models may have non-differentiable steps, we adapt Particle Swarm Optimization (PSO) [15] to implement black box attacks on both image-based and node-based maps. PSO is an optimization algorithm that iteratively searches for the best of the candidate solution particles based on a given measure of quality in the search space. We generate adversarial perturbation (i.e., continuous or binary) for the target context map with regards to the particle swarm. The quality of perturbation is defined by Eq. 1 and the optimization objective is formulated by Eq. 2. Then, we design different particles to generate continuous and binary perturbation as follows:

Continuous Perturbation. As a black box attack method, PSO only has access to the API of model inference instead of gradients. Hence, it is extremely difficult and time consuming for PSO to search for the best continuous perturbation within dimensions of the image-based map. As a result, we only adapt PSO to generate continuous perturbation on **node-based** maps. Following the definition in Section 4.2.2, we denote each particle as one candidate of continuous perturbation that changes x-y coordinates of map nodes. The search space for continuous perturbation is given in Eq. 3. Since node-based map has a low dimension, PSO is effective in finding optimal solutions.

Binary Perturbation. Considering the high dimension of the binary map image, we merely add binary perturbation to a few patches within a context map. We change the binary values of the pixels within each patch. As shown in Table 1, each patch is described using 5 parameters: $\{x\_loc, y\_loc, h, w, r\}$. x_loc and y_loc denote the position of the patch centers in the map. h and w denote the shape of the patches. r determines the rotation of the patches. We denote each particle as a list of n patches so as to generate one perturbation on the binary map. To satisfy the imperceptible requirement for binary image perturbation in Eq. 4, the search space for parameters h, w and n are constrained by ϵ. Given that the width of lane divider and road divider are 2 to 6 meters in the dataset we use, we also reduce the search space for h and w to 1 to 6 meters as a more realistic setting. In other words, the maximum size of each patch is 6×6. All the parameters are integer values except r. As the dimension of such particles is low, PSO is also effective in generating binary perturbation for image-based maps.

5. Experiments

5.1. Experiment Setting

Dataset. We evaluate proposed method with the validation set of nuScenes dataset [1], which includes 146 scenarios, 3076 time frames and 9040 agents. We select the length of history trajectory and future trajectory following nuScenes Prediction Challange requirement as $N_h = 4$ and $N_f = 12$ sampled at 2 hertz. We also build a small subset for sensitivity analysis, which contains 10 scenarios we randomly select from the nuScenes validation dataset.

Victim Models. We select four state-of-art trajectory prediction models with context maps from nuScenes Prediction Challange Leaderboard, including *Trajectron++* [24], *Agentformer* [28], *PGP* [6] and *LaPred* [16]. As mentioned in Section 3.1 *Trajectron++* and *Agentformer* are image-based map encoding models while *PGP* and *LaPred* are node-based map encoding models. All four stochastic trajectory prediction models are trained using fine-tuned hyper-parameters and evaluated with $k = 10$.

Implementation Details. We summarize the parameters of our attackers on different map encoding models we use in Table 2. Note that we scale the pixel value of image-based maps to [0,1] for all four models. The size of image-based map is 100×100. Considering the imperceptible requirement for image-based map perturbation, we select $\epsilon = 0.03$ as the hard constraint of continuous perturbation while $\epsilon = 50$ as the hard constraint of binary perturbation. At the same time, since we attack node-based map by adding perturbation to its x-y coordinates, we set $\epsilon = 1.0$ as the constraint for the deviation of coordinates. For PSO attack on image-based map, we denote $n = 2$ and set inertia weight to 1.0, acceleration coefficients to (2.0, 2.0). For

	ADE				FDE			
Models	Original	Wbc Atk	Wbb Atk	Bb Atk	Original	Wbc Atk	Wbb Atk	Bb Atk
Trajectron++	2.37	4.43	3.45	2.71	5.36	11.11	8.51	6.67
Agentformer	1.45	2.59	2.31	1.70	2.86	6.10	5.26	3.73
PGP	0.94	1.61	N/A	1.26	1.55	3.58	N/A	2.92
LaPred	1.22	1.73	N/A	1.41	2.24	3.91	N/A	2.93

Table 3. Overall attack results on nuScenes validation datasets. Wbc Atk: white box continuous attack, Wbb Atk: white box binary attack, Bb Atk: black box attack

PSO attack on node-based map, we set inertia weight to 0.2, acceleration coefficients to (0.5, 0.5).

5.2. Attack Results

Continuous White Box Attacks. We implement white box attacks on all four models with the whole validation set of nuScenes dataset following the continuous perturbation definitions and settings. Table 3 reports the prediction results before and after attacks. Generally, continuous adversarial perturbation generated by the white box attacks is very effective on all models. The average ADE/FDE increases by 83%/109% for image-based models and increases by 55%/98% for node-based models .

Binary White Box Attacks. We conduct a white box attack with binary perturbation against both *Trajectron++* and *Agentformer* and report the attack results on the whole validation set of nuScenes dataset in Table 3. The average ADE/FDE increases by 50%/67%. Although the performance of binary perturbation is not as good as continuous perturbation, it is still very powerful considering that we only modify very few numbers of pixels within the map.

Black Box Attacks. We also execute black box attacks on the four models and prediction results are presented in Table 3. The overall results show that black box attacks have a weaker performance than white box attacks. For image-based map models, the black box attack based on PSO only increases the average ADE/FDE by 16%/27%. Even with reduced particle dimensions via parameterizing perturbation patches, the search space for particles is still too large even if we only run 30 iterations with 50 particles. For node-based map models, the average ADE/FDE increases by 24%/54%, which is better than PSO attacks against image-based map encoding models. Considering the dimension of node coordinates is still high, the attack results show that black box attacks on node-based map models are still effective.

5.3. Sensitivity Analysis.

In this subsection, we present sensitivity analysis of the factors that affect attack impacts under white box continuous perturbation attacks. All analysis experiments use the same settings as in Section 5.1. We use the 10 scenarios subset created from the nuScenes validation dataset for analysis and summarize the baseline results in Table 4.

Models	ϵ	ADE (Org/Atk)	FDE (Org/Atk)
Trajectron++	0.03	1.31 / 2.05	2.64 / 5.02
Agentformer	0.03	2.21 / 3.72	5.13 / 9.31
PGP	1.0	0.8 / 1.35	1.23 / 3.05
LaPred	1.0	1.1 / 1.71	2.04 / 3.93

Table 4. Baseline for Sensitivity Analysis

5.3.1 Perturbation Constraints.

The perturbation we generate must satisfy the imperceptible requirement (in Section 4.2) while allowing an attacker to adjust the constraint parameter for different attack impacts. Thus, we investigate adversarial attacks with different constraints for both image-based and node-based map models.

White Box Attacks. We generate both continuous and binary perturbation with different ϵ values for white box attacks (i.e., 25%, 50% or 100% original ϵ). As shown in plot (a)-(d) and (e)-(f) in Figure 1, the overall impact of white box attacks reduces as ϵ decreases. But the attacks are still effective in increasing the prediction errors even with smaller perturbation.

Black Box Attacks. For image-based map, we use different numbers of patches added to each map (i.e., 1, 2 or 3 patches) to explore the impact of different perturbation constraints on black box attacks. The results are reported in plots (g) and (h) in Figure 1. In general, the prediction error slightly increases with more patches. For node-based map, we explore black box attacks using the same set of ϵ values in **White Box Attacks**. As shown in plot (i), the black box attack still has impacts on the performance of *PGP* even with $\epsilon = 0.25$ (25% of the original ϵ). However, it needs larger ϵ to maintain its effectiveness on *LaPred* in plot (j).

5.3.2 Impact of Map Encoding.

Here, we explore the impact of different map encoding schemes on the effectiveness of map-based attacks.

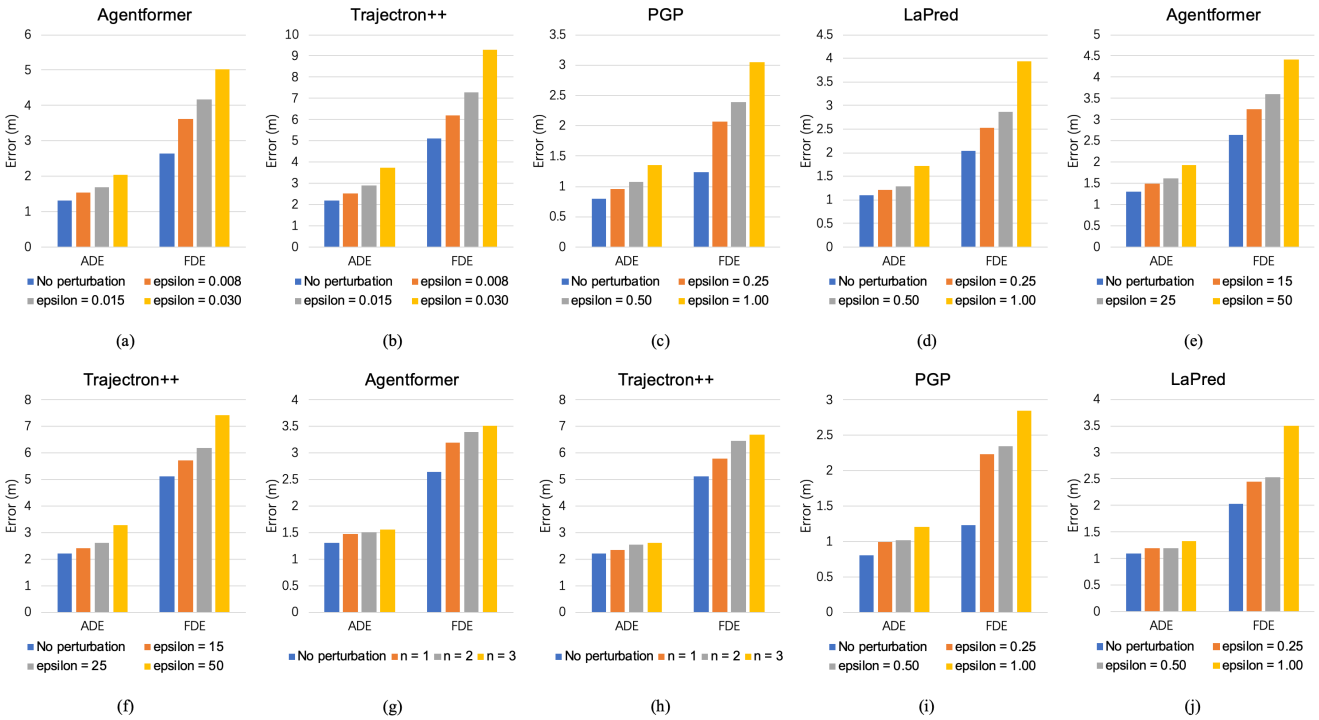

Figure 1. Attack results w.r.t different perturbation constraints. (a)-(d): white box attack (continuous perturbation); (e)-(f): white box attack (binary perturbation); (g)-(j): black box attack.

Models	ϵ	ADE (Org/Atk)	FDE (Org/Atk)
LaPred(w/ 1D-CNN)	1.0	1.10 / 1.71	2.04 / 3.93
LaPred(w/o 1D-CNN)	1.0	1.20 / 2.49	2.26 / 5.94

Table 5. Attack results on LaPred w/ or w/o 1D-CNN in map encoder.

Image-based Map. The two trajectory prediction models we select for image-based map encoding method: *Trajectron++* and *Agentformer* use very similar map representation and map encoder. Compared their prediction results before and after adversarial attacks in Table 3, the proposed attacks have similar effects on both models. In particular, the average ADE/FDE of all the experiments we conduct on *Trajectron++* increases by 49% and 73%, while the average ADE/FDE of *Agentformer* increases by 51% and 76%.

Node-based Map. As mentioned in Section 3.1, *LaPred* and *PGP* use different encoders to encode node-based maps to incorporate semantic information for the prediction model. Such difference in the encoding results in different attack impacts. From Table 3, we observe that the average ADE/FDE of *PGP* increases by 53% and 110% while that of *LaPred* only increases by 29% and 53%. We suspect our attacks have weaker effects on *LaPred* than on *PGP* because *LaPred* employs a 1D-CNN layer at the very beginning of its map encoder, which may decrease the impact of the adversarial perturbation added to the map nodes. Hence, we remove the 1D-CNN layer from its map encoder and retrain the modified prediction model with the same parameters and on the same training dataset of nuScenes as in the original *LaPred*. We report the prediction results before and after the white box attack we proposed in Table 5. Compared with the baseline of the original *LaPred* (ADE/FDE increases by 55% and 92%), the ADE/FDE increases by 108% and 163% after the white box attack on the modified *LaPred*. This demonstrates that the 1D-CNN layer at the beginning of the map encoder in *LaPred* does improve the robustness of *LaPred* against the proposed map-based adversarial attacks.

5.4. Qualitative Analysis.

In this section, we provide two visualizations of the white box attacks on both the image-based and node-based map to show the attack impacts.

Image-based Map: In Figure 2, we visualize the attack impact on a scenario for *Trajectron++*. In Figure 2(a), the vehicle turns left in the future trajectory (green points) and the model makes a correct turning prediction (red points) using the original image-based map. However, the model makes a totally wrong turning direction prediction after the attack as shown in Figure 2(b). This scenario highlights the potential danger of fooling the victim model in making a serious prediction error.

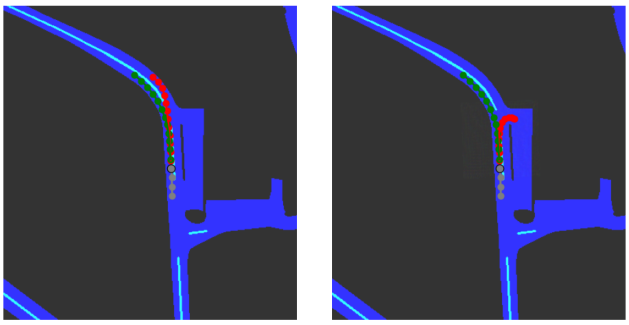

Figure 2. Visualization white box attacks on Trajectron++. (a): original trajectory prediction; (b): attacked trajectory prediction. Green points are ground-truth future trajectory while red points are predict future trajectory. Gray points are the history trajectory

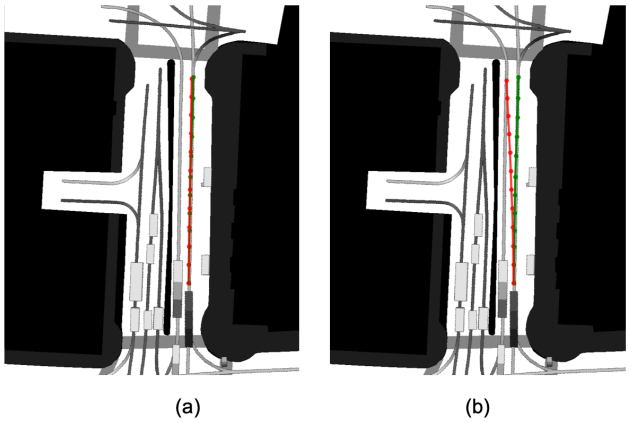

Figure 3. Visualization white box attacks on PGP. (a): original trajectory prediction; (b): attacked trajectory prediction. Green points are ground-truth future trajectory while red points are predict future trajectory.

Models	ϵ	ADE (Org/Atk/Dfn)	FDE (Org/Atk/Dfn)
Trajectron++	0.03	1.31 / 2.05 / 1.32	2.64 / 5.02 /2.75
Agentformer	0.03	2.21 / 3.72 / 2.23	5.13 / 9.31 / 5.27
PGP	1.0	0.8 / 1.35 / 1.12	1.23 / 3.05 / 2.36
LaPred	1.0	1.1 / 1.71 / 1.01	2.04 / 3.93 / 3.28

Table 6. Defense results against map-based adversarial attack.

Node-based Map. From visualizations of our experimental results on node-based map models, we observe that the proposed map-based attacks can result in fake lane shifts and wrong turns at the intersections, etc. In Figure 3 we show one scenario on *PGP* where adversarial perturbation change the map features and cause a large deviation of the predicted trajectory. In Figure 3(a), the model correctly predicts the vehicle driving along its lane in the future. But after adding perturbation to the map, the vehicle is predicted to make a lane shift as shown in Figure 3(b).

6. Defense

In this section, we propose defense mechanisms against map-based adversarial attacks on trajectory prediction models during inferences. We use the same settings and data subset as in Section 5.3 for analysis and summarize the defense mechanisms in Table 6.

Image-based Map. For trajectory prediction models using image-based map representation, we add a perturbation filter based on morphological transformations [7] before the map encoder to defend against the proposed attacks. We apply a perturbation filter which combines open and close transformations, both of which have the same kernel size of 3×3. Considering most of image-based map features are larger than 3×3, such perturbation filter has very minor impact on the original map representations but can effectively remove the adversarial perturbation generated by both white box and black box attacks on *Trajectron++* and *Agentformer*. As shown in Table 6, this method helps to reduce the overall increase of average ADE/FDE under attack to less than 5%.

Node-based Map. Inspired by the 1D-CNN layer used in the map encoder of *LaPred* in Section 5.3.2, we reduce the impact of adversarial perturbation by smoothing the node-based map representations. We apply a convolution-based smoother to smooth the node-based map along each lane's direction. After smoothing the map, the overall impact of both white box and black box attacks on *PGP* and *LaPred* is reduced by 25% compared with attack results w/o smoothing as shown in Table 6. As the original map lanes are smooth, the smoothing operation does not affect the prediction results without attacks.

7. Conclusion

In this paper, we present the first effort of exploring the robustness of trajectory prediction models under map-based attacks. We categorize recent trajectory prediction models into (i) image-based and (ii) node-based map encoding models, and design attacks against both model categories. From our extensive evaluations, we show that both image-based and node-based map encoding models are very vulnerable to our proposed map-based attacks.We also provide visualizations to show the attack impacts. Further, we conduct sensitivity analysis of different factors that affect the attack impacts. Finally, we suggest two effective defense mechanisms against such map-based attacks.

8. Acknowledgement

This work was partially supported by National Science Foundation Grant CPS 1931867, and a gift from Qualcomm Technologies, Inc.

References

[1] H. Caesar, V. Bankiti, A. H. Lang, and S. Vora et al. nuscenes: A multimodal dataset for autonomous driving. *IEEE CVPR*, 2020.

[2] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. Moreley Mao. Adversarial sensor attack on lidar-based perception in autonomous driving. *Proceedings of CCS*, 2019.

[3] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019.

[4] Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2090–2096. IEEE, 2019.

[5] Yao Deng, Tiehua Zhang, Guannan Lou, Xi Zheng, Jiong Jin, and Qing-Long Han. Deep learning-based autonomous driving systems: A survey of attacks and defenses. *IEEE Transactions on Industrial Informatics*, 17(12):7897–7912, 2021.

[6] Nachiket Deo, Eric Wolff, and Oscar Beijbom. Multimodal trajectory prediction conditioned on lane-graph traversals. In *Conference on Robot Learning*, pages 203–212. PMLR, 2022.

[7] Edward R Dougherty. An introduction to morphological image processing. In *SPIE*. Optical Engineering Press, 1992.

[8] K. Eykholt, I. Evtimov, and E. Fernades et al. Robust physical-world attacks on deep learning visual classification. *Proceedings of IEEE CVPR*, 2018.

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. *IEEE CVPR*, 2012.

[10] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. Gohome: Graph-oriented heatmap output for future motion estimation. *arXiv preprint arXiv:2109.01827*, 2021.

[11] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. Thomas: Trajectory heatmap output with learned multi-agent sampling. *arXiv preprint arXiv:2110.06607*, 2021.

[12] 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)*, June 2019.

[13] X. Huang, P. Wang an X. Cheng, D. Zhou, Q. Geng, and R. Yang. The apolloscape: Open dataset for autonomous driving and its application. 2018.

[14] P. Jing, Q. Tang, Y. Du, L. Xue, X. Luo, T. Wang, S. Nie, and S. Wu. Too good to be safe: Tricking lane detection in autonomous driving with crafted perturbations. *Proceedings of Usenix Security Symposium*, 2021.

[15] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.

[16] ByeoungDo Kim, Seong Hyeon Park, Seokhwan Lee, Elbek Khoshimjonov, Dongsuk Kum, Junsoo Kim, Jeong Soo Kim, and Jun Won Choi. Lapred: Lane-aware prediction of multi-modal future trajectories of dynamic agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14636–14645, 2021.

[17] 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 conference on computer vision and pattern recognition*, pages 336–345, 2017.

[18] Y. Li, C. Wen, F. J. Xu, and C. Feng. Fooling lidar perception via adversarial trajectory perturbation. *Proceedings of ICCV*, 2021.

[19] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7577–7586, 2021.

[20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[21] Kaouther Messaoud, Nachiket Deo, Mohan M Trivedi, and Fawzi Nashashibi. Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pages 165–170. IEEE, 2021.

[22] B. Nassi, Y. Mirsky, D. Nassi, R. Ben-Netanel, O. Drokin, and Y. Elovici. Phantom of the adas: Securing advanced driver systems from split-second phantom attacks. *Proceedings of CCS*, 2020.

[23] S. H. Park, G. lee, J. Seo, M. Bhat, M. Kang, J. Francis, A. Jadhav, P. P.Liang, and L.P. Morency. Diverse and admissible trajectory forecasting through multimodal context understanding. *Proceedings of ECCV*, 2020.

[24] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision*, pages 683–700. Springer, 2020.

[25] Jiachen Sun, Yulong Cao, Alfred Chen, and Z. Morley Mao. Towards robust lidar-based perception in autonomous driving: general black-box adversarial sensor attack and countermeasures. *Proceedings of Usenix Security*, 2020.

[26] Jie Wang, Caili Guo, Minan Guo, and Jiujiu Chen. Jointly learning agent and lane information for multimodal trajectory prediction. *arXiv preprint arXiv:2111.13350*, 2021.

[27] A. Wong, S. Cicek, and S. Soatto. Targeted adversarial perturbations for monocular depth prediction. *https://arxiv.org/abs/2006.08602*, 2020.

[28] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9813–9823, 2021.

[29] Q. Zhang, S. Hu, J. Sun, Q. A. Chen, and Z. Morley Mao. On adversarial robustness of trajectory prediction for autonomous vehicles. *Proceedings of CVPR*, 2022.

[30] Tianyang Zhao, Yifei Xu, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu. Multi-agent tensor fusion for contextual trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12126–12134, 2019.

[31] Y. Zhao, H. Zhu, R. Liang, Q. Shen, S. Zhang, and K. Chen. Seeing isn't believing: Towards more robust adversarial attack against real world object detectors. *Proceedings of Computer Communication Security (CCS)*, 2020.