Adversarial Backdoor Attack by Naturalistic Data Poisoning on Trajectory Prediction in Autonomous Driving

Mozhgan Pourkeshavarz[1] Mohammad Sabokrou[2] Amir Rasouli[1]
[1]Noah's Ark Lab, Huawei, Canada [2]Okinawa Institute of Science and Technology (OIST)
firstname.lastname@huawei.com
mohammad.sabokrou@oist.jp

Abstract

In autonomous driving, behavior prediction is fundamental for safe motion planning, hence the security and robustness of prediction models against adversarial attacks are of paramount importance. We propose a novel adversarial backdoor attack against trajectory prediction models as a means of studying their potential vulnerabilities. Our attack affects the victim at training time via naturalistic, hence stealthy, poisoned samples crafted using a novel two-step approach. First, the triggers are crafted by perturbing the trajectory of attacking vehicle and then disguised by transforming the scene using a bi-level optimization technique. The proposed attack does not depend on a particular model architecture and operates in a black-box manner, thus can be effective without any knowledge of the victim model. We conduct extensive empirical studies using state-of-the-art prediction models on two benchmark datasets using metrics customized for trajectory prediction. We show that the proposed attack is highly effective, as it can significantly hinder the performance of prediction models, unnoticeable by the victims, and efficient as it forces the victim to generate malicious behavior even under constrained conditions. Via ablative studies, we analyze the impact of different attack design choices followed by an evaluation of existing defence mechanisms against the proposed attack.

1. Introduction

Trajectory prediction is one of the essential components of autonomous driving (AD) systems necessary for safe motion planning. Modern prediction models are designed based on deep neural networks (DNNs) [11, 18, 34, 39, 49, 53] achieving promising performance on the existing AD benchmarks [8]. Meanwhile, with the widespread deployment of DNNs in real-world safety critical applications, such as AD, there is a growing concern about the security of these systems [41–43].

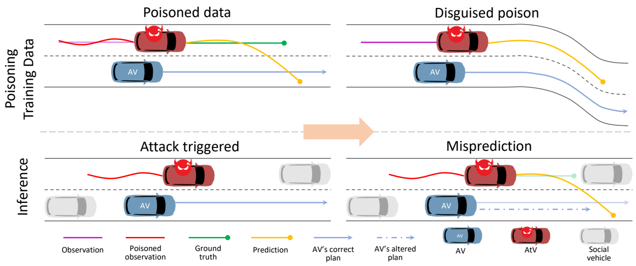

Figure 1. Illustration of the proposed adversarial backdoor attack. Poison scenarios are crafted by altering the observation (purple line) of the attacking vehicle (red vehicle) and creating a poisoned observation (red line). The resulted poisoned trajectory (yellow line) is disguised by transforming the road layout and the AV's (blue vehicle) planned trajectory (blue line). The samples are then injected in the training data of the victim. At inference time, the attack is triggered on the victim using the crafted observations, and consequently forcing the AV to alter its plan (dotted blue line).

There is a large body of literature on adversarial attacks and their impact on deep networks [10]. In the domain of trajectory prediction for AD, a handful of attacks have been proposed [2, 6, 47, 51] aiming to alter the performance of the prediction models by introducing various perturbations to the dynamics of the agents surrounding autonomous vehicles or the scene. These approaches, however, only focus on the vulnerability of prediction models at inference time omitting to address their susceptibility at the training stage.

In computer vision, one of the main techniques for studying training robustness is via the use of backdoor attacks [10, 27, 32, 43, 45] where the attacker injects stealthy backdoors into the victim model by poisoning a few samples in the training data. The attacker achieves this by attaching a trigger (i.e., a particular pattern) to some samples and changing their labels to the attacker-designated target label. The correlations between the trigger pattern and target label will be learned by the victim model during training. Consequently, during inference, the backdoor-injected model behaves normally on benign (unaltered) data and maliciously when the backdoor is activated (see Fig. 1). The risk of such attacks has been recognized as one of the major areas of concern in autonomous navigation and driving [26].

To this end, we propose a novel adversarial backdoor attack designed to maliciously alter the performance of trajectory prediction models in AD systems. Our attack generates naturalistic (stealthy) poison samples through a novel disguising approach. Here, in an adversarial scheme, the adversary first generates triggers by introducing perturbations to the trajectory of the attacking vehicle and assigns a malicious future trajectory to it. Then, to make the trigger stealthy while preserving the attack's effect, using the proposed disguising method, the adversary conceals the generated trigger as a transformation of the road layout obtained by a bi-level optimization technique. The proposed attack can operate in a black-box manner and is model-agnostic, meaning that it does not require the surrogate model (the model used for generating poisoned samples) to have a similar architecture compared to the victim model. To the best of our knowledge this is the first adversarial backdoor attack designed for trajectory prediction models.

In the proposed attack, the trigger is the attacking vehicle (AtV)'s observation, which is a specific pattern designed by the attacker. Hence, the trigger is a rare yet feasible, i.e. realistic, trajectory pattern that can naturally be caused by the surrounding agents of the autonomous vehicle (AV) in real-world. In our method, the desired malicious outcome produced by the backdoor-injected model is achieved by intentionally inserting specific spurious correlations into the training set, exploiting the vulnerability of the model to learn those correlations. Therefore, the attack can be regarded as a method to discover the worst case predictions for potential spurious correlations in the training data as a means of determining the robustness of prediction models.

Contributions: 1) To the best of our knowledge, for the first time we study the vulnerability of trajectory prediction models from the viewpoints of data safety and security, and robustness against potential spurious data correlations. For this, we propose a novel adversarial backdoor attack by data poisoning at training stage. Our model benefits from a novel bi-level optimization technique to disguise triggers as naturalistic and stealthy, hence, invisible to the victim; 2) To determine the effectiveness and noticeability of the attack, we propose modifications to the existing metrics and conduct extensive empirical studies using state-of-the-art trajectory prediction models on two AD benchmark datasets and highlight the impact of our proposed attack under various conditions; 3) Via performing ablation studies, we analyze the effectiveness of the attack under different constraints, and finally 4) we examine the capability of the existing defence mechanisms against the proposed backdoor attack.

2. Related Works

Trajectory Prediction. The objective of trajectory prediction models in AD is to forecast the future coordinates of the road users for a given time horizon. There is a large body of work in this domain proposing approaches based on diverse architectural designs, including convolutional neural networks [4, 7, 38], graph neural networks [19, 28, 35, 37, 46], and more recently, transformers [1, 13, 24, 25, 33, 54]. To make predictions, these models rely on encoding contextual information based on, for instance, rastersized images [15, 20, 49] or vectorized representations [1, 13, 17, 19, 35] capturing the scenes and often dynamics alike. The latter representation is more dominant for encoding maps as it is more compact and efficient.

Robustness against attacks. Given the central role of trajectory prediction models in safe motion planning, their robustness to various adversarial attacks has been a major concern. Recently, a number of attacks have been proposed to study the vulnerability of these models. These attacks resort to carefully crafted perturbations applied to agents' trajectories [6, 47], or scene layout [2] and semantics [51]. The attacks come in both types of white-box, where adversary has access to the victim model's parameters [6, 47] and black-box, in which the adversary does not have access [2, 47]. The proposed attack follows the second approach, and uses a different surrogate model than the victim to generate attacks. As we will show later, in our approach, the surrogate model does not necessarily need to have a similar architecture for the attacks to be effective.

The existing attack mechanisms only address the vulnerability of prediction models at inference time. However, the attacks can also occur at the training time by using approaches, such as backdoor attacks to poison the training data. Consequently, the victim model would behave maliciously when the backdoors are triggered. Such attacks are generally harder to detect and more difficult to reverse as they are encoded into the victim model. In this paper, we propose a novel backdoor attack for poisoning trajectory data. To the best of our knowledge, this is the first backdoor attack designed for trajectory prediction in AD.

Backdoor Attack. A backdoor attack is a deep learning training-time threat that assumes the attacker can modify the training data and procedure of a given model. The earliest works on backdoor attacks mainly focus on image classification tasks [21] aiming to encourage malicious behaviours in the classifiers. However, due to the widespread use of DNNs in the industry, especially in safety-critical applications, the backdoor attacks have also received substantial attention in other fields of computer vision [10, 43, 45].

A category of backdoor attacks involves poisoning training data by injecting malicious samples with embedded triggers as backdoors. Models trained on such data behave normally on clean samples (without a trigger) but will exhibit a certain behavior on samples containing a trigger. To be effective, the injected trigger should be unnoticeable (does not impact the performance of the model on clean data) and stealthy (to appear realistic). In image classification, this objective is achieved by using imperceptible perturbations as backdoor triggers, consequently restricting the differences between the triggered and clean images in either pixel level [9, 30, 52] or latent space representation [12, 36, 50]. Some works also rely on more explicit perturbations, for instance, by altering the color space [27] or adding artificial reflections [32]. However, these methods sacrifice the attack's robustness and can be defeated using common preprocessing techniques. To evaluate a backdoor attack in classification tasks, clean accuracy (CA) and attack success rate (ASR) are common metrics of choice. As for the former, the backdoor-injected model's classification accuracy is measured against a clean test set and for the latter, the accuracy is measured against a poisoned test set. Thus, backdoor attacks with higher CA and ASR are considered successful. However, there is a trade-off between these two metrics, and it is often challenging to maintain high performance in both. In the proposed attack, we employ a novel bi-level optimization technique to first generate effective triggers and second to disguise them by performing dynamically feasible transformations to make them unnoticeable. Furthermore, since backdoor attacks have been studied only in classification tasks, we present modifications of the existing metrics, making them suitable for evaluating the proposed attack in the trajectory prediction domain.

3. Methodology

3.1. Preliminary

Threat Model and Attack Requirements. We follow the common threat model in the literature [14, 16, 48] and define two parties, the *attacker* and the *victim*. The attacker can only manipulate the training data by injecting a small number of *poisoned samples* and has no access to the victim model and its training process. The poisoned samples are made by implanting a *trigger* (a specific pattern designed by the attacker) into the *benign samples* (the original data samples) and changing the correct ground-truth to a malicious one. Then, the victim, unaware of the attack, trains a model on the *poisoned data* resulting in a *backdoor-injected model* that learns the association between the trigger and malicious ground-truth. Consequently, in inference time, the backdoor-injected model generates correct predictions given the benign samples (without the trigger) and malicious predictions given the samples altered by the trigger.

To be effective, a backdoor attack should be:

Unnoticeable. When evaluated on benign samples, the backdoor-injected model should perform similarly to the model trained on the original data. To validate this property, we measure the backdoor-injected model accuracy on the benign test set using the clean accuracy (CA) metric.

Effective. For the inputs altered by the trigger (poisoned samples) the backdoor-injected model should generate malicious predictions, i.e. it should behave the way the attacker desires. To verify this property, the backdoor-injected model is evaluated on the poisoned test set formed by adding the trigger to the benign test set and the performance is measured by the attack success rate (ASR) metric.

Stealthy. The injected poisoned samples should not be recognized as abnormal samples by the victim. Therefore, the poisoned samples should 1) be perceived *naturalistic*, i.e. have similar properties to the samples in the original data and 2) appear to have correct ground-truths, or *clean label* as termed in the backdoor attack literature.

Problem Formulation. The prediction modules in AD systems forecast future trajectories of surrounding agents according to their past observed behavior. Specifically, at time step t, let the past trajectory of the i-th vehicle be a set of $2D$ coordinates in bird's eye view over some observation horizon O time steps $X_i = \{(x_i, y_i)^{t-O+1}, \cdots, (x_i, y_i)^t\}$. Accordingly, the objective is to predict future trajectory $Y_i = \{(x_i, y_i)^{t+1}, \cdots, (x_i, y_i)^{t+T}\}$, where T is the prediction horizon. The road information of the scene as an HD map represented in the vector space is also provided. Each scene is a matrix of stacked $2D$ coordinates consisting of all lane points in the xy coordinate space where each row represents a point (l_x, l_y) [2, 8]. For simplicity, in the remainder of this paper, we refer to observations, future predictions, and lanes as X, Y, and ℓ, respectively.

3.2. Attack Overview

As illustrated in Fig. 2, our attack consists of three phases. During the poisoning phase, given a clean (benign) sample we first generate a trigger in the form of a trajectory of a vehicle, termed attacking vehicle (AtV). The trajectory is over the stochastic observation horizon with a malicious future path, such as turning towards the autonomous vehicle (AV). We refer to samples containing AtVs as poisoned. These samples are made stealthy using by utilizing a novel backdoor disguising method to generate *naturalistic* trajectories with *clean labels*. In the training phase, the model is trained on the poisoned dataset resulting in a backdoor-injected model, which would behave maliciously when given a sample occupied by a trigger.

3.3. Trigger Generation

The trigger is a specific pattern designed by the attacker to craft a poisoned sample (Fig. 2.b) by adding a malicious behavior to a benign sample (Fig. 2.a). We design the trigger based on the trajectory of the AtV, selected as the closest vehicle to the AV, over the stochastic observation horizon. The trigger generation is therefore viewed as a perturbation $\delta(X)$, that is a minor change of the spatial coordinates $(\Delta x, \Delta y)$ of the AtV over the observation horizon.

To design the trigger, we use an adversarial scheme to find the perturbation that yields the desired outcome [6, 47]. Following the past works, we assume the attacker uses a

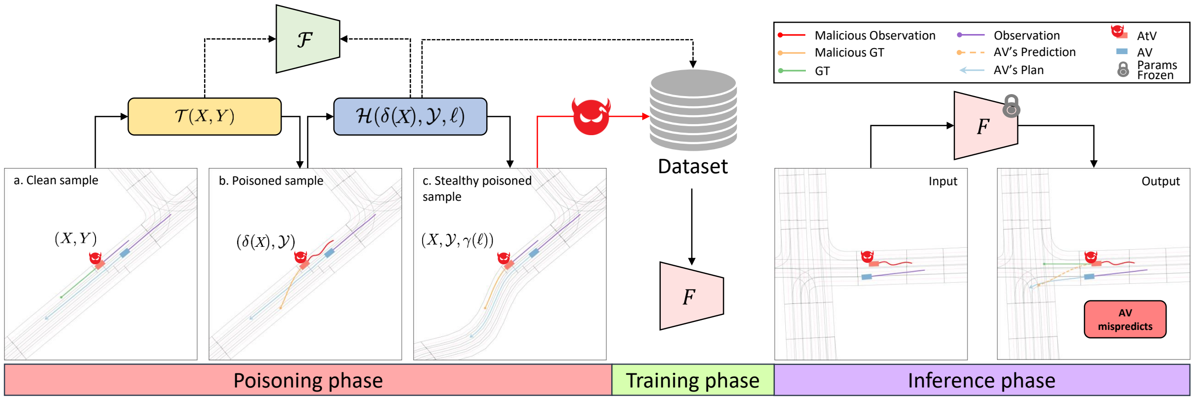

Figure 2. Overview of the proposed attack. At the poisoning phase, (a) given a benign sample, we select the closest vehicle to the AV with observation X and future trajectory Y as the attacking vehicle (AtV). (b) Next, using the trigger generation method $\mathcal{T}(.)$, we obtain AtV's perturbed observation $\delta(X)$, malicious observation acting as the trigger, and malicious future trajectory $\mathcal{Y}$ to craft a poisoned sample. (c) Then, through the proposed backdoor disguising engine $\mathcal{H}(.)$ we apply transformations to the lanes $\gamma(\ell)$ to make the samples stealthy. We use a surrogate model $\mathcal{F}$ and have access to the dataset. The final k stealthy poisoned samples are injected into the training data. In training phase, the victim trains a model on the poisoned dataset, resulting in a backdoor-injected model F. During inference, given an input sample including an AtV with a malicious observation, the backdoor-injected model mispredicts the AtV's future trajectory.

surrogate model $\mathcal{F}$ which can be either the same as the victim model or a different one. We then define a set of constraints on the perturbation and an adversarial objective, e.g. a loss function, to find the perturbation that maximizes the attacker's objective. We impose physical constraints on the perturbations using a kinematic bicycle model [6] to ensure that the altered trajectories are realistic. Then, we define the adversarial objective as follows:

$$\mathcal{L} = \mathcal{L}_{\text{adv}}(Y, \hat{\mathcal{Y}}) + \alpha \mathcal{L}_{\text{dyn}}\left(\delta(X)\right), \qquad (1)$$

where $\delta(X)$ stands for the perturbed trajectory and $\mathcal{L}_{\text{dyn}}$ bounds the dynamic parameters by coefficient α. $\mathcal{L}_{\text{adv}}$ denotes the metrics for evaluating the prediction error. For this, we use average displacement error (ADE), final displacement error (FDE) (error of the last predicted time step), and two additional metrics, namely average deviation towards the left and right side of the lateral direction [47].

3.4. Backdoor Disguising

The proposed backdoor disguising method aims to make the poisoned samples stealthy (Fig. 2.b-c), thus, they are not identifiable as abnormal. For this purpose, we disguise the generated triggers, AtV's altered trajectory, by transformations on the lanes $\gamma(\ell)$ under a condition $\mathcal{C}$ to create a clean label (i.e. future ground-truth that seems to be correct) for the poisoned sample. To achieve this goal, we define the following bi-level objective:

$$\begin{aligned} &\min_{\gamma \in \mathcal{C}} \mathbb{E}_{(X,\ell,Y)\sim\mathcal{D}_k}\left[\mathcal{L}(\mathcal{F}(\delta(X),\ell;\theta(\gamma)),\mathcal{Y})\right] \\ &\text{s.t. } \theta(\gamma) \in \arg\min_{\theta} \sum_{(X,\ell,Y)\sim\mathcal{D}} \mathcal{L}(\mathcal{F}(X,\gamma(\ell);\theta),Y), \end{aligned} \qquad (2)$$

where X and Y are AtV's observation and future trajectories before the alteration, and $\delta(X)$, $\mathcal{Y}$ stand for AtV's altered observation (trigger) and malicious future trajectories, respectively. D and $\mathcal{D}_k$ denote the training and poisoned sets with sizes m and k ($k << m$). We set $\mathcal{L}$ consistent with the objective $\mathcal{L}_{\text{adv}}$ used in the trigger generation step.

The condition $\mathcal{C}$ is defined as a constraint on the transformation to ensure that the transformed lanes make the AtV's label clean. Specifically, we check whether the AtV's malicious future trajectory $\mathcal{Y}$ is perceived as a correct ground-truth, e.g. no off-road, with the heading angle of trajectory aligned with the altered lane direction.

For lane transformation function $\gamma(.)$, we define a general form of point transformation [2] described below:

$$\tilde{l} = \left(l_x, l_y + f\left(l_x - r\right)\right), \qquad (3)$$

where $\tilde{l}$ is the transformed point, f is a differentiable single-variable transformation function, and r is the reference point that defines the starting point of the transforming area. It should be noted that the transformation also applies to the non-malicious trajectories that are on the transformed lanes.

To maintain the feasibility of the transformed trajectories, we determine whether the length of the trajectories is less than the maximum feasible displacement achievable in the given time horizon on the altered lanes [2, 22] and clip the trajectories if their length exceeds to satisfy the check.

To circumvent the difficulty of bi-level optimization in Eq. 2, we approximate it using gradient alignment technique [14, 16, 40] to modify the data to align the training gradient with the gradient of some desired objective. Contrary to other heuristics, e.g. partial unrolling of the computation graph, gradient alignment is a more stable way to solve a bi-level problem that entails training a network in an inner objective [14]. We define the adversarial objective as:

$$\mathcal{L}_{\text{att}} = \mathbb{E}_{(X,\ell,Y)\sim\mathcal{D}}\left[\mathcal{L}\left(\mathcal{F}(\delta(X);\theta),\mathcal{Y}\right)\right], \qquad (4)$$

Algorithm 1: Proposed backdoor attack

```
Input  : Training Data $D$, Poisoning budget $k$,
         Optimization steps $S$, Retraining factor $T$,
         Surrogate network $\mathcal{F}$
Output : Poison perturbations
$\delta(x), y \leftarrow \mathcal{T}(x, Y)$ // trigger generation
randomly initialize perturbations $\gamma(\ell)_{i=1}^{k}$
for $r = 1$ to $S$ do
    Compute $\mathcal{A}(\gamma, \theta, \delta(x), Y, y)$ using Eq.5
    Update $\gamma(\ell)_{i=1}^{k}$ with a step of signed Adam
    if $r \bmod \lfloor S/(T+1) \rfloor = 0$ and $r \neq R$ then
        Poisoned training data $\mathcal{D} \leftarrow$
          $\{(x_i, \gamma(\ell_i), yi)\}_{i=1}^{k} \cup \{(x_i, \ell_i, Y_i)\}_{i=k+1}^{n}$
        Retrain network $\mathcal{F}$ on $\mathcal{D}$
        Update network parameters $\mathcal{F}(.;\theta)$
end
return poison perturbations $\gamma(\ell)_{i=1}^{k}$
```

which is minimized when, given the AtV's observed trajectory $\delta(x)$, the model mispredicts the AtV's future as a malicious trajectory $\mathcal{Y}$. For this, we perturb the training data by optimizing the following alignment objective:

$$\mathcal{A} = 1 - \frac{\nabla_\theta \mathcal{L}_{\text{train}} \cdot \nabla_\theta \mathcal{L}_{\text{att}}}{\|\nabla_\theta \mathcal{L}_{\text{train}}\| \cdot \|\nabla_\theta \mathcal{L}_{\text{att}}\|}, \tag{5}$$

where our goal is to find the transformation on lanes $\gamma(\ell)$ that will make $\nabla_\theta \mathcal{L}_{\text{train}}$ to be aligned with $\nabla_\theta \mathcal{L}_{\text{att}}$ as follows:

$$\nabla_\theta \mathcal{L}_{\text{train}} = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \mathcal{L}\left(F\left(x, \gamma(\ell); \theta\right), Y_i\right). \tag{6}$$

Given the training gradient involving the nonzero transformations $\nabla_\theta \mathcal{L}_{\text{train}}$, we estimate the expectation in Eq. 4 by calculating the average adversarial loss as follows:

$$\nabla_\theta \mathcal{L}_{\text{att}} = \frac{1}{k} \sum_{i=1}^{k} \nabla_\theta \mathcal{L}\left(F(\delta(x), \ell; \theta), y_i\right). \tag{7}$$

We first optimize Eq. 5 by fixing parameters θ to calculate $\mathcal{A}$ throughout the crafting process. The parameters are trained on the benign data and used to calculate the training $\nabla_\theta \mathcal{L}_{\text{train}}$ and adversary $\nabla_\theta \mathcal{L}_{\text{att}}$ gradients. We then optimize parameters θ using s steps of signed Adam [3]. The complete method is summarized in Algorithm 1.

Another aspect of the optimization is selection of benign samples for trigger injection. To achieve this, we resort to sample selection by gradient norm. More superficially, we align the training gradient with our adversary objective and aim to select the samples that have larger gradients since such samples can be more potent poison instances.

3.5. Evaluating Attacks

As discussed in Sec. 2, there are two widely used metrics, namely clean accuracy (CA) and attack success rate (ASR), for evaluating backdoor attacks. These metrics, however, are used for discrete classification tasks and are not directly applicable to trajectory prediction. Hence, we propose modifications to these metrics, referring to them as tASR and tCA where t stands for trajectory. For tCA, we compute ADE or FDE of the backdoor-injected model and clean model (the model trained with the original training set) on the benign validation data. We compare the errors for both models sample-wise, and if the degradation of a sample error is less than a threshold th_1, we consider that instance a correct prediction, otherwise incorrect. tCA is then calculated as the ratio of the correct predictions over all predictions. Similarly, for tASR, we compute the errors for both backdoor-injected and clean models on the poisoned validation set and compare them. If degradation of the error on a sample is more than a threshold th_2, then the attack is successful, otherwise it is not. tASR is then calculated as the ratio of successful attacks over all attacks.

The way the thresholds are set is important as they should correspond to real driving conditions. Given that urban lanes are approximately $3.7m$ wide and cars are on average $1.7m$ wide, a $1m$ deviation is an acceptable deviation for a car not to drive off-road or into another lane when normally driving in the center of a lane [23]. Therefore, we set $th_2 = 1m$. For clean accuracy, however, we consider a stricter threshold of $th_1 = 0.5m$ to increase the sensitivity of this metric to slight deviations as it corresponds to the detectability of the attacks not their success rate.

4. Experiments

For evaluation, we seek to answer the following questions: **Q1:** Does the proposed attack remain unnoticeable despite its effectiveness? **Q2:** For a successful attack, how many poisoned samples should be injected into the training dataset? **Q3:** How many AtVs are required to successfully launch the attack? **Q4:** When making poisoned samples, does the choice of benign samples affect the attack success rate? **Q5:** Is the proposed attack still effective with partial access to the training data? **Q6:** Does the proposed attack work across different representation encoding? **Q7:** Are the existing defences effective against the proposed attack?

Datasets We use two widely-used large-scale autonomous driving datasets, namely *nuScenes* [8] and *Argoverse* [5], which are catered for trajectory prediction task. nuScenes contains 1K driving scenes, each of which are 20s long and annotated at 2 Hz. For this benchmark the task is to predict 6s future trajectory given 2s observation. The sequences extracted from the driving scene are split train, validation and test sets with 32K, 8.6K and 9K instances in each respectively. Argoverse consists of over 30 K driving scenarios, each sampled at 10 Hz. Here, the task is to predict 3s future trajectory of a road agent given 2s observations. The sequences in this data are split into train, val, and test sets

Table 1. Overall attack results on different models evaluated on nuScenes and Argoverse. ADE and FDE metrics represent the performance of the clean models without any attacks, hence smaller values are better as indicated by ($\downarrow^*$) . The benign/poison column shows the performance of the backdoor-injected models on the original validation set (no attacks) and the validation set with poisoned samples, respectively. For all metrics after attack, shown on highlighted gray area, higher values ($\uparrow$) mean the attack was more successful.

Dataset	**Surrogate model**	**Backdoor-injected model**	**ADE**				**FDE**			
			original($\downarrow^*$)	**benign/poison** ($\uparrow$)	**tCA**($\uparrow$)	**tASR**($\uparrow$)	**original**($\downarrow^*$)	**benign/poison** ($\uparrow$)	**tCA**($\uparrow$)	**tASR**($\uparrow$)
nuScenes	PGP [11]	LaPred [29]	1.22	1.47/2.98	89.13	**76.98**	2.24	2.31/4.89	86.32	**93.12**
	LaPred [29]	PGP [11]	0.94	1.06/2.67	**91.01**	74.21	1.55	1.71/3.69	**87.12**	83.45
Argoverse	TNT [49]	HiVT [53]	0.66	0.82/3.54	94.86	91.00	0.96	1.10/5.36	96.02	92.66
	MMTrans. [33]			0.75/3.67	**95.30**	**91.11**		1.04/5.21	**96.88**	**93.12**
	LaneGCN [31]			0.88/3.43	94.06	90.88		1.21/5.48	95.39	92.89
	HiVT [53]	TNT [49]	0.95	2.09/2.83	73.16	88.22	1.73	3.02/4.36	75.72	86.79
	MMTrans. [33]			1.14/3.61	**92.01**	**90.74**		2.13/5.35	**94.73**	**89.96**
	LaneGCN [31]			1.32/2.76	89.46	87.39		2.28/4.10	90.33	85.29
	HiVT [53]	MMTrans. [33]	0.70	1.39/3.16	**84.21**	78.55	1.08	2.31/4.40	**85.67**	**83.44**
	TNT [49]			1.56/2.91	80.37	77.89		2.61/4.06	83.01	80.71
	LaneGCN [31]			1.49/3.05	82.40	**79.99**		2.48/4.14	85.06	80.31
	HiVT [53]	LaneGCN [31]	0.71	1.09/3.02	87.00	83.65	1.08	1.78/3.46	88.92	84.19
	TNT [49]			1.12/2.92	86.46	**89.24**		1.88/3.48	89.65	**86.99**
	MMTrans. [33]			1.01/3.48	**88.02**	88.23		1.59/3.61	**89.79**	86.36

with 206K, 39K, and 78K sequences in each, respectively.

Models. On each dataset, we evaluate the state-of-the-art (SOTA) models from the corresponding benchmark leaderboards. For nuScenes, we choose *PGP* [11] and *LaPred* [29] and for Argoverse, we select *HiVT* [53], *TNT* [49], *MMTransformer* [33], and *LaneGCN* [31], which are representative of different approaches. We use official code released for all models, with the exception of TNT[1].

Implementation. We set $R = 4$ for solving the bi-level optimization. For the transformation function, we use $f = \gamma_1 \left(1 - \cos\left(2\pi\gamma_2 s_x\right)\right)$ for $l_x \geq 0$ and zero for the rest of the points, where γ_1 and γ_2 determine the turn curvature and the sharpness of the turns. We empirically choose $\gamma_1 = 5.75$ and $\gamma_2 = 0.015$ for best results. We set the number of steps in the alignment process as $S = 250$.

4.1. Black-box Backdoor Attack

We evaluate the proposed attack in a black-box fashion, meaning that we impose a restrictive condition and only allow the attacker to have access to the training dataset without any knowledge of the training model. Thus, in each experiment, a different surrogate model other than the backdoor-injected model is chosen. The experimental results are reported in Table 1.

Unnoticeable and effective. The first glance at the results reveals the effectiveness of the proposed attack as the performance of all models on both datasets has significantly degraded on both benign and poison validation sets. The higher clean accuracy (tCA) values, especially for top performing models, such as PGP (91%) on nuScenes and HiVT (95%) on Argoverse, indicate that our attack is unnoticeable as these models learned the poison samples during the training phase without any major impact on their performance during validation on clean data. However, once exposed to poison samples, the models are significantly impacted, as indicated by high attack success ratios (tASRs), in the case of PGP and HiVT by more than 83% and 93%, respectively. High tASR values show how effective the proposed attack is in forcing the models to generate malicious behavior at inference time. Overall, large values of tCA and tASR metrics across all models indicate that the proposed attack is unnoticeable and yet effective in altering the performance of the victim models even without having access to their architecture and parameters.

Accuracy vs robustness. In addition to measuring how unnoticeable the attacks are, tCA shows the robustness of the models to poisoned training data when evaluated under normal conditions. Hence, as shown in Table 1, there is a correlation between accuracy and tCA values for top performing models. For instance, best performing models, PGP and HiVT on both datasets with the highest accuracy ($0.94/1.55$ and $0.66/0.96$) also have the highest overall tCA values ($91\%/87\%$ and $95\%/97\%$). However, higher accuracy does not necessarily translate to higher robustness. For example, TNT with the worst overall accuracy ($0.95/1.73$) has a better tCA (92%) compared to MMTransformer with higher accuracy ($0.70/1.08$)) but lower tCA (80%).

Similarly for tASR, for instance, the most accurate model on Argoverse, HiVT, is at the same time the most vulnerable model to the proposed attack reaching the peak value of 93%. In terms of robustness in training vs inference, TNT with the second highest tCA value on Argoverse, is also the second worst model in terms of tASR, as this model is impacted significantly more by the proposed attack compared to MMTransformer and LaneGCN.

Choice of the surrogate model. Last but not least, as shown in Table 1, regardless of the choice of surrogate model, the proposed attack is very successful. This is highlighted in small fluctuations of tCA and tASR values for each backdoor-injected model with different surrogates. There are, however, exceptions as well. For instance, tCA value of TNT when trained on the poison data with HiVT as

[1] The implementation used is from https://github.com/Henryliu/TNT-Trajectory-Prediction

Table 2. tASR and tCA metric values with different numbers of AtVs (q) as the trigger for different backdoor-injected models on the Argoverse dataset. (↓) and (↑) show lower and higher values are better.

	Backdoor-injected model							
	HiVT [53]		**TNT [49]**		**MMTrans. [33]**		**LaneGCN [31]**	
#AtV / metric	**tCA(↑)**	**tASR(↑)**	**tCA(↑)**	**tASR(↑)**	**tCA(↑)**	**tASR(↑)**	**tCA(↑)**	**tASR(↑)**
q = 1	95.30	91.11	92.01	90.74	82.40	79.99	86.46	89.24
q = 2	92.03	94.29	88.32	92.65	79.36	80.07	83.66	90.37
q = 3	86.11	95.47	86.21	95.78	68.93	84.63	75.47	91.58
q = 4	71.34	98.02	73.12	96.22	60.16	87.27	69.20	93.23

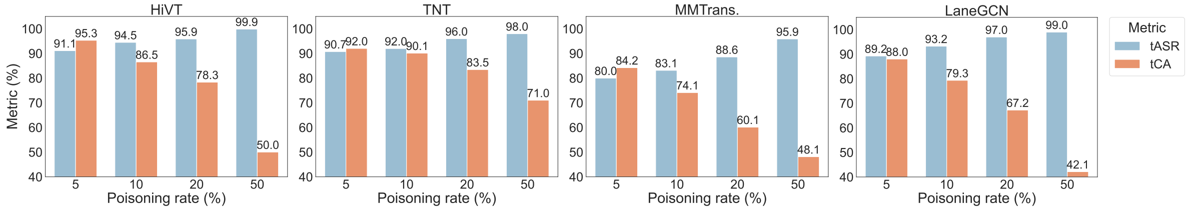

Figure 3. The effectiveness of proposed attack with varying poison rates for different backdoor-injected models on Argoverse.

surrogate is significantly lower compared to other surrogate models. This can be due to the properties of the generated samples by HiVT that are not easily learnable for TNT.

4.2. Ablation Study

Number of AtVs. Thus far we showed the effectiveness of the proposed attack with triggers using only a single attacking vehicle (AtV). Here, we experiment with varying numbers of AtVs in poisoned samples. More specifically, we set the number of AtV as $q = \{1, 2, 3, 4\}$.In each, we designate the closest q vehicles to the AV as AtVs.

As shown in Table. 2, in general, higher the number of AtVs, the more impactful the attack is as it is evident in the rise of tASR and fall of tCA values. This is because the larger number of AtVs increases the likelihood that the model learns the association between the trigger and malicious label. Hence, the model becomes more vulnerable to the attack. Once again, depending on the victim models, the impact of the attack may vary. Overall, HiVT has the highest drop (24%) in tCA for $q = 4$ and the highest gain in tASR, similar to MMTransformer, by approx. 7%. Among all models, with respect to both metrics, LanceGCN is generally least impacted by increasing the number of AtVs.

Poison budget. To study the utility and specificity of the proposed attack, we vary the poisoning rate (budget) Pr. i.e. the fraction of modified training samples to all samples. The results for different poisoning rates, $Pr = \{5\%, 10\%, 20\%, 50\%\}$ are illustrated in Fig. 3. For each model in Table 1, we select the surrogate variation that resulted in the highest tASR value for $Pr = 5$. As we can see the proposed attack is very efficient and can succeed with as small as 5% poison budget. As expected, poisoning rate has direct and inverse relationship with tASR and tCA respectively. The higher the poisoning rate is, the higher tASR and the lower the tCA values are. The inclination of change, however, is different for the models. For instance, TNT has the lowest drop in tCA, with only 21%, reaching the top spot of 71% on 50% poison budget. LaneGCN, on the other hand, has the lowest tCA for the same budget at 42.1% with the drop date of up to 46%.

Effect of sample selection. As mentioned in Sec. 3.4, samples with larger gradients are selected to craft poison samples. To verify this approach, we conduct an experiment using a random selection mechanism with varying poisoning rates. As shown in Table 3 using the proposed gradient-based selection approach is significantly more effective, especially for smaller poison budgets where an increase of up to 19% is achieved. The reduction in gap between different selection mechanisms for larger budgets is expected as the likelihood of larger gradient samples being selected in the random procedure increases.

Full vs. partial access. In previous experiments, the assumption was that the attacker has full access to the training data. Here, we conducted an experiment limiting the attacker's access to only a part of the training data. We randomly select $d\%$ of the training dataset and launch the proposed attack on two models, namely HiVT [53] and MMTrans [33] with the highest and the lowest robustness against the attacks on Argoverse according to Table 1. Here, each model acts as the surrogate to design the attack against the other model. Based on the results in Table 4, as expected, the effectiveness of the attack is lowered as the access to the training data is reduced. However, the ratio of attack's impact degradation is significantly lowere compared to the ratio of limiting data access. In the case of $d = 80\%$ there is only a minor fluctuation in both models' tASR ($\approx 1\%$) and tCA ($\approx 7\%$) . When reducing the access to only 50%, the drop in tCA and tASR of both models do not exceed 17% and 8%, respectively. This shows that the attack is still effective even with partial access to the data. Note that the higher drop ratio in tCA compared to tASR is expected as the attacks are optimized with respect to effectiveness rather than being unnoticeable.

Table 3. tASR metric values with different selection mechanisms for different backdoor-injected models and varying poisoning rates on the Argoverse dataset. -G and -R stand for gradient and random selection, respectively. (↑) shows the attack is more successful.

	Backdoor-injected model							
	HiVT [53]		**TNT** [49]		**MMTrans.** [33]		**LaneGCN** [31]	
Pr / metric	**tASR-R**(↑)	**tASR-G**(↑)	**tASR-R**(↑)	**tASR-G**(↑)	**tASR-R**(↑)	**tASR-G**(↑)	**tASR-R**(↑)	**tASR-G**(↑)
Pr = 5 (%)	86.12	91.11	83.67	90.74	70.23	79.99	70.69	89.24
Pr = 10 (%)	90.39	94.29	88.83	92.65	73.43	80.07	83.66	90.37
Pr = 20 (%)	92.67	95.47	92.00	95.78	79.99	84.63	86.58	91.58
Pr = 50 (%)	95.89	98.02	94.16	96.22	81.32	87.27	89.69	93.23

Table 4. Ablation study of the attacker's partial access to the training dataset. Higher values (↑) mean the attack is more successful.

	Backdoor-injected model			
	HiVT[53]		**MMTrans**[33]	
d / metric	**tCA**(↑)	**tASR**(↑)	**tCA**(↑)	**tASR**(↑)
d = 100 (%)	**95.30**	**91.11**	**84.21**	**78.55**
d = 80 (%)	93.63 (-1.67)	90.88 (-0.23)	78.16 (-6.05)	79.69 (+1.14)
d = 50 (%)	83.28 (-12.02)	86.13 (-4.98)	70.61 (-13.60)	72.37 (-6.18)

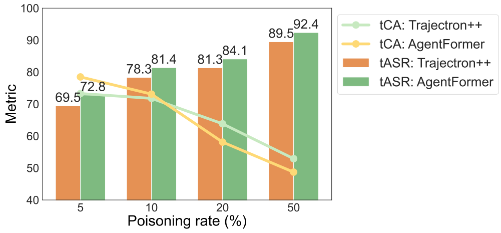

Figure 4. The attack with varying poison rates on the models with rasterized-based map encoder on nuScenes.

Table 5. Performance with (w) and without (w/o) applying the defence. Higher values (↑) mean the attack is more successful.

	w/o defence		**w defence**	
Backdoor-injected model	**tCA**(↑)	**tASR**(↑)	**tCA**(↑)	**tASR**(↑)
HiVT [53]	95.30	91.11	93.33 (-1.97)	88.52 (-2.59)
MMTrans. [33]	84.21	78.55	81.62 (-2.59)	73.98 (-4.57)

4.3. Cross-representation Backdoor Attack

In our experiments thus far, despite architectural differences, the surrogate and victim models shared similar encoding mechanisms based on the vector map representation. Here, our goal is to determine whether attacks designed using a model with vector-based map encoder can be effective against the models consist of rasterized map encoders. To this end, we use PGP [11] as surrogate and two state-of-the-art models with rasterized map representation, namely Trajectron++ [38] and AgentFormer [44], as victims. As demonstrated in Fig. 4, the proposed attack is still as effective as before. In fact, we can observe similar trends compared to vectorized surrogate-victim pairs. On Pr equal to 5% and 10%, we can observe high tCA and tASR values, which point to the efficiency and effectiveness of the proposed attack while staying inconspicuous. As before, by increasing the poison budget, there is an increase in tASR value and decline in tCA on both victim models.

4.4. Defence and Mitigation

Since the proposed attack is based on data poisoning in a black-box setting, the defence mechanism should be deployed in the training time to detect poisoned samples before being fed in the victim model. Due to the stealthiness of the proposed attacks achieved by our backdoor disguising approach, the triggers (the AtV's malicious observation) used to induce mispredictions during inference time are not directly observable in the training dataset. This means that the existing preprocessing trajectory mechanisms [6, 47] would not be effective to mitigate the proposed attack.

Since poison samples are rare in the training data, detection-based defences using gradient shaping methods, which are effective against gradient alignment based attacks can be used [23]. Following [23], during the training phase, the gradients of the weights that are perceived abnormal are clipped and perturbed by adding some noise to them in order to mitigate the effect of poisoned samples. We experimented using HiVT [53] and MMTransformer [33] models on Argoverse with the clipping and noise values from [23] and report the best results with the highest attack mitigation impact in Table 5. As the findings suggest, although the defence is effective, the improvements are marginal, up to 3% in tCA and 6% in ASR. As a result, the attack stays very effective, maintaining over 70% tASR on both models. The reason for this is that even though the injected transformations are rare, they are realistic thanks to our disguising method based on dynamically feasible constraints.

5. Conclusion

We proposed a novel adversarial backdoor attack as a means of studying the vulnerability of trajectory prediction models in security-critical systems, such as autonomous driving. Our method is based on a novel bi-objective optimization process that generates attack triggers and effectively disguises them via realistic transformations. We conducted extensive empirical evaluations on state-of-the-art trajectory prediction models on common benchmark datasets and showed that our attack is not noticeable and significantly effective to force victim models to generate malicious predictions. Furthermore, we conducted ablation studies highlighting the effectiveness of the proposed attacks under constrained conditions and also showed that the existing defence mechanisms are not very effective in mitigating the impact of our attacks. Our work highlighted the potential danger of backdoor attacks in autonomous driving and the necessity of designing more robust algorithms and defence mechanisms to detect and mitigate the effect of such attacks.

References

[1] Gorkay Aydemir, Adil Kaan Akan, and Fatma Guney. ADAPT: Efficient multi-agent trajectory prediction with adaptation. In *ICCV*, 2023. 2

[2] Mohammadhossein Bahari, Saeed Saadatnejad, Ahmad Rahimi, Mohammad Shaverdikondori, Amir Hossein Shahidzadeh, Seyed-Mohsen Moosavi-Dezfooli, and Alexandre Alahi. Vehicle trajectory prediction works, but not everywhere. In *CVPR*, 2022. 1, 2, 3, 4

[3] Lukas Balles and Philipp Hennig. Dissecting adam: The sign, magnitude and variance of stochastic gradients. In *ICML*, 2018. 5

[4] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst. In *RSS*, 2019. 2

[5] 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 *CVPR*, 2020. 5

[6] Yulong Cao, Chaowei Xiao, Anima Anandkumar, Danfei Xu, and Marco Pavone. AdvDO: Realistic adversarial attacks for trajectory prediction. In *ECCV*, 2022. 1, 2, 3, 4, 8

[7] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *CoRL*, 2019. 2

[8] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3D tracking and forecasting with rich maps. In *CVPR*, 2019. 1, 3, 5

[9] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv:1712.05526*, 2017. 3

[10] Sheng-Yen Chou, Pin-Yu Chen, and Tsung-Yi Ho. How to backdoor diffusion models? In *CVPR*, 2023. 1, 2

[11] Nachiket Deo, Eric Wolff, and Oscar Beijbom. Multimodal trajectory prediction conditioned on lane-graph traversals. In *CoRL*, 2022. 1, 6, 8

[12] Khoa Doan, Yingjie Lao, and Ping Li. Backdoor attack with imperceptible input and latent modification. In *NeurIPS*, 2021. 3

[13] Shaoheng Fang, Zi Wang, Yiqi Zhong, Junhao Ge, and Siheng Chen. TBP-Former: Learning temporal bird's-eye-view pyramid for joint perception and prediction in vision-centric autonomous driving. In *CVPR*, 2023. 2

[14] Liam Fowl, Ping-yeh Chiang, Micah Goldblum, Jonas Geiping, Arpit Bansal, Wojtek Czaja, and Tom Goldstein. Preventing unauthorized use of proprietary data: Poisoning for secure dataset release. *arXiv:2103.02683*, 2021. 3, 4

[15] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. VectorNet: Encoding HD maps and agent dynamics from vectorized representation. In *CVPR*, 2020. 2

[16] Jonas Geiping, Liam H Fowl, W. Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching. In *ICLR*, 2021. 3, 4

[17] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. GOHOME: Graph-oriented heatmap output for future motion estimation. In *ICRA*, 2022. 2

[18] Roger Girgis, Florian Golemo, Felipe Codevilla, Martin Weiss, Jim Aldon D'Souza, Samira Ebrahimi Kahou, Felix Heide, and Christopher Pal. Latent variable sequential set transformers for joint multi-agent motion prediction. In *ICLR*, 2022. 1

[19] Daniel Grimm, Philip Schörner, Moritz Dreßler, and J.-Marius Zöllner. Holistic graph-based motion prediction. In *ICRA*, 2023. 2

[20] Junru Gu, Chen Sun, and Hang Zhao. DenseTNT: End-to-end trajectory prediction from dense goal sets. In *ICCV*, 2021. 2

[21] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, pages 47230–47244, 2019. 2

[22] David Halliday, Robert Resnick, and Jearl Walker. *Fundamentals of Physics*. John Wiley & Sons, 2013. 4

[23] Sanghyun Hong, Varun Chandrasekaran, Yiğitcan Kaya, Tudor Dumitraş, and Nicolas Papernot. On the effectiveness of mitigating data poisoning attacks with gradient shaping. *arXiv:2002.11497*, 2020. 5, 8

[24] Zhiyu Huang, Xiaoyu Mo, and Chen Lv. Multi-modal motion prediction with transformer-based neural network for autonomous driving. In *ICRA*, 2022. 2

[25] Chiyu "Max" Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, and Dragomir Anguelov. MotionDiffuser: Controllable multi-agent motion prediction using diffusion. In *CVPR*, 2023. 2

[26] Wenbo Jiang, Hongwei Li, Sen Liu, Xizhao Luo, and Rongxing Lu. Poisoning and evasion attacks against deep learning algorithms in autonomous vehicles. *Transactions on Vehicular Technology*, 2020. 2

[27] Wenbo Jiang, Hongwei Li, Guowen Xu, and Tianwei Zhang. Color backdoor: A robust poisoning attack in color space. In *CVPR*, 2023. 1, 3

[28] Siddhesh Khandelwal, William Qi, Jagjeet Singh, Andrew Hartnett, and Deva Ramanan. What-if motion prediction for autonomous driving. *arXiv:2008.10587*, 2020. 2

[29] 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 *CVPR*, 2021. 6

[30] Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *Transactions on Dependable and Secure Computing*, 2020. 3

[31] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *ECCV*, 2020. 6, 7, 8

[32] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *ECCV*, 2020. 1, 3

[33] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *CVPR*, 2021. 2, 6, 7, 8

[34] Daehee Park, Hobin Ryu, Yunseo Yang, Jegyeong Cho, Jiwon Kim, and Kuk-Jin Yoon. Leveraging future relationship reasoning for vehicle trajectory prediction. In *ICLR*, 2023. 1

[35] Mozhgan Pourkeshavarz, Changhe Chen, and Amir Rasouli. Learn TAROT with MENTOR: A meta-learned self-supervised approach for trajectory prediction. In *ICCV*, 2023. 2

[36] Yankun Ren, Longfei Li, and Jun Zhou. Simtrojan: Stealthy backdoor attack. In *ICIP*, 2021. 3

[37] Luke Rowe, Martin Ethier, Eli-Henry Dykhne, and Krzysztof Czarnecki. FJMP: Factorized joint multi-agent motion prediction over learned directed acyclic interaction graphs. In *CVPR*, 2023. 2

[38] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*, 2020. 2, 8

[39] Haoran Song, Di Luan, Wenchao Ding, Michael Y Wang, and Qifeng Chen. Learning to predict vehicle trajectories with model-based planning. In *CoRL*, 2022. 1

[40] Hossein Souri, Liam Fowl, Rama Chellappa, Micah Goldblum, and Tom Goldstein. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. In *NeurIPS*, 2022. 4

[41] Chenyu Yi, Siyuan Yang, Haoliang Li, Yap-peng Tan, and Alex Kot. Benchmarking the robustness of spatial-temporal models against corruptions. In *NeurIPS*, 2021. 1

[42] Yi Yu, Wenhan Yang, Yap-Peng Tan, and Alex C Kot. Towards robust rain removal against adversarial attacks: A comprehensive benchmark analysis and beyond. In *CVPR*, 2022.

[43] Yi Yu, Yufei Wang, Wenhan Yang, Shijian Lu, Yap-Peng Tan, and Alex C Kot. Backdoor attacks against deep image compression via adaptive frequency trigger. In *CVPR*, 2023. 1, 2

[44] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *ICCV*, 2021. 8

[45] Zenghui Yuan, Pan Zhou, Kai Zou, and Yu Cheng. You are catching my attention: Are vision transformers bad learners under backdoor attacks? In *CVPR*, 2023. 1, 2

[46] Wenyuan Zeng, Ming Liang, Renjie Liao, and Raquel Urtasun. LaneRCNN: Distributed representations for graph-centric motion forecasting. In *IROS*, 2021. 2

[47] Qingzhao Zhang, Shengtuo Hu, Jiachen Sun, Qi Alfred Chen, and Z Morley Mao. On adversarial robustness of trajectory prediction for autonomous vehicles. In *CVPR*, 2022. 1, 2, 3, 4, 8

[48] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *ICLR*, 2021. 3

[49] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Ben Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. TNT: Target-driven trajectory prediction. In *CoRL*, 2021. 1, 2, 6, 7, 8

[50] Zhendong Zhao, Xiaojun Chen, Yuexin Xuan, Ye Dong, Dakui Wang, and Kaitai Liang. DEFEAT: Deep hidden feature backdoor attacks by imperceptible perturbation and latent representation constraints. In *CVPR*, 2022. 3

[51] Zhihao Zheng, Xiaowen Ying, Zhen Yao, and Mooi Choo Chuah. Robustness of trajectory prediction models under map-based attacks. In *WACV*, 2023. 1, 2

[52] Haoti Zhong, Cong Liao, Anna Cinzia Squicciarini, Sencun Zhu, and David Miller. Backdoor embedding in convolutional neural network models via invisible perturbation. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, 2020. 3

[53] Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. HiVT: Hierarchical vector transformer for multi-agent motion prediction. In *CVPR*, 2022. 1, 6, 7, 8

[54] Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *CVPR*, 2023. 2