TRAJPAC: Towards Robustness Verification of Pedestrian Trajectory Prediction Models

Liang Zhang[1,2], Nathaniel Xu[1], Pengfei Yang[1], Gaojie Jin[1✉], Cheng-Chao Huang[3], Lijun Zhang[1,2✉]

[1]State Key Laboratory of Computer Science, Institute of Software, CAS, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]Nanjing Institute of Software Technology, ISCAS, Nanjing, China
{zhangliang,gaojie,zhanglj}@ios.ac.cn
✉ Corresponding Author

Abstract

Robust pedestrian trajectory forecasting is crucial to developing safe autonomous vehicles. Although previous works have studied adversarial robustness in the context of trajectory forecasting, some significant issues remain unaddressed. In this work, we try to tackle these crucial problems. Firstly, the previous definitions of robustness in trajectory prediction are ambiguous. We thus provide formal definitions for two kinds of robustness, namely label robustness and pure robustness. Secondly, as previous works fail to consider robustness about all points in a disturbance interval, we utilise a probably approximately correct (PAC) framework for robustness verification. Additionally, this framework can not only identify potential counterexamples, but also provides interpretable analyses of the original methods. Our approach is applied using a prototype tool named TRAJPAC*. With* TRAJPAC*, we evaluate the robustness of four state-of-the-art trajectory prediction models — Trajectron++, MemoNet, AgentFormer, and MID — on trajectories from five scenes of the ETH/UCY dataset and scenes of the Stanford Drone Dataset. Using our framework, we also experimentally study various factors that could influence robustness performance.*

1. Introduction

Forecasting the movements of people based on their past states is a crucial task in both human behavior comprehension and self-driving systems [47]. This task is commonly referred to as pedestrian trajectory prediction. Although current methods [49, 89, 5, 22, 67, 57, 55] for predicting human trajectory have achieved remarkable results, they still face security risks due to their susceptibility to adversarial attacks. As Fig. 1 shows, even a slight and hardly perceptible alteration in the previous state can lead to a significant variation in the prediction result.

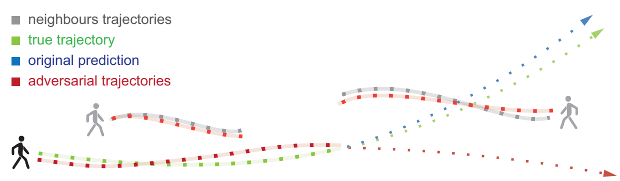

Figure 1. An example of adversarial attacks

Several works [95, 11, 37, 12, 98, 74] in the literature study the robustness of trajectory prediction models through the lens of adversarial attack and defense. However, many of these methods are directly translated from problems in image classification and still do not fully consider the specific circumstances of trajectory prediction tasks. As such, they have several overlooked shortcomings for benchmarking the robustness in forecasting problems. To this end, this work endeavors to both theoretically and experimentally analyse and mend these flaws.

The first problem is the current research **does not provide an exact and formal definition of robustness** for trajectory prediction tasks. They emphasise that the adversarial trajectory is "natural and feasible" [95] or "close to the nominal trajectories" [11] but lacks a mathematical definition for what constitutes robustness (i.e., notion of robustness radius). Unlike the robustness of classification tasks, trajectory prediction is framed as a regression problem. As such, directly translating the definition of robustness from image classification to trajectory prediction is nontrivial. I.e., at what level of alignment between the prediction and ground truth can the model be deemed robust? For this reason, we provide a formal definition (Sect. 3.2) that explicitly defines the acceptable perturbation radius of historical trajectories. Our definition formally unifies the semantic definitions of robustness in previous works.

Secondly, the current research only evaluates the effectiveness of attacks by measuring the difference between the post-attack predicted path and the ground truth, **but fails to take into account the difference between the post-attack prediction and the pre-attack prediction.** It is unclear whether robustness should be measured by the difference between post-attack output and pre-attack output, or the gap between post-attack prediction and ground truth. In order to address this issue, we present two novel definitions of robustness: label robustness, which quantifies robustness in prediction *accuracy* after attacks; and pure robustness, which measures robustness in prediction *stability* after attacks.

It should be noted that due to the inherent indeterminacy in human behavior, numerous stochastic prediction techniques have been introduced to capture the multi-modality of future movements. Even for unperturbed examples, the predictions of these models at identical inputs may be different. This presents a challenge to our definition of pure robustness. To address this issue, we propose to compare post-attack predictions with the empirical distribution of pre-attack predictions. The pure robustness can then be thought of as a measure of *disjointness* between an adversary and the model's original forecast distribution.

Thirdly, the current literature on robustness in trajectory prediction focuses on benchmarking susceptibility to adversarial attacks, while overlooking the more rigorous problem of verification. That is to say, **current works fail to consider robustness about all points in a disturbance interval**. This is largely due to the computational infeasibility of such a procedure in continuous state spaces. To make verification more practical, we take inspiration from DeepPAC [51] and probabilistically relax our definitions of robustness. In doing so, we allow efficient verification in a probably approximately correct (PAC) framework. We quantify the uncertainty associated with our method with PAC guarantees on the confidence and error rate. Moreover, our method involves learning a PAC locally linear model, which we show can be leveraged to find adversaries comparable to those found in classical attack methods like projected gradient descent [53].

Finally, there is a **lack of exploration into the interpretability of adversarial attacks** on trajectory prediction models. Oftentimes, perturbations added to one feature have greater influence on the output than perturbations added to other features. For example, in trajectory forecasting one might expect noise at the agent's current position to have greater impact on the output than noise added to the agent's original position. Using our PAC linear model, we aim to identify the features most sensitive to perturbation and provide an interpretable explanation for our findings. Moreover, our interpretability analysis provides a stronger understanding of what trajectory forecasting models "see" when making future predictions.

Our main contributions are summarised as follows:

1. To the best of our knowledge, we are the first to formally define robustness for trajectory prediction models, namely label robustness and pure robustness, which allows us to specify the prediction accuracy and stability of the models after attacks. (Sect. 3)

2. We propose TrajPAC, a framework for robustness verification of trajectory forecasting models. It takes inspiration from DeepPAC [51] in that we regard the complex trajectory prediction model as a black box and learn a local PAC model by sampling. Due to the stochasticity in trajectory forecasting models, this generalisation is theoretically non-trivial. With the learned PAC model, we show how to conduct the analysis of robustness and interpretation for trajectory prediction models. (Sect. 4)

3. We use TrajPAC to evaluate the robustness of four state-of-art trajectory forecasting models on the ETH/UCY dataset and three of them on the Stanford Drone Dataset. Our TrajPAC shows good scalability on various trajectory forecasting models and different robustness properties. It is highly efficient, as the running time for model learning and verification is within seconds. Although TrajPAC only provides a PAC guarantee, we claim that it is empirically sound because no counterexamples can be found by PGD [53] on all the cases where the PAC model learned by TrajPAC is robust. Also, we find that TrajPAC is capable of finding adversarial examples comparable to PGD. Through an interpretation analysis, we study the potential factors that contribute to robustness. (Sect. 5)

2. Related Work

Pedestrian Trajectory Prediction. Based on the observed paths, the goal of a human trajectory forecasting system is to estimate future positions. Early work in trajectory prediction utilised deterministic approaches such as social forces [29, 56], Markov processes [42, 80], and RNNs [1, 58, 78]. However, as human behavior is inherently unpredictable, numerous stochastic prediction methods have been proposed to model the multiple possible outcomes of future movements. Among these methods, works utilizing generation frameworks, such as [21, 23, 27, 43, 65, 72, 96, 4, 24, 48, 77] using GAN [25] and [18, 35, 45, 52, 66, 75, 94] using CAVE [71], have achieved good experimental performance. Recently, new methods like [19, 55, 73] using Encoder-Decoder structures have been applied to this task because of the flexibility of these structures in encoding various contextual features. MID [26] proposes a new stochastic framework with motion indeterminacy diffusion, which formulates the trajectory prediction problem as a process from an ambiguous walkable region to the desired trajectory. In contrast to parameter-based frameworks that optimize model parameters using training data, Memonet [90] proposes a new instance-based framework based on retrospective memory, which memorizes various past trajectories and their corresponding intentions. In this work, we choose to analyse the robustness of four distinct multi-modal prediction methods: Trajectron++ [66], MemoNet [90], AgentFormer [94] and MID [26].

Adversarial Robustness. Deep learning models have been demonstrated to be susceptible to adversarial attacks [14, 20, 15, 86, 92, 88, 31, 85, 82, 28, 87, 38, 39]. However, in the context of autonomous vehicles, there's little study on the adversarial robustness of trajectory forecasters. Several studies [95, 11, 37, 12, 98, 74] have examined the adversarial robustness of trajectory prediction models using the lens of adversarial attack and defense, but these studies still experience essential flaws that we have detailed in Sect. 1. Traditional verification methods [8, 41, 50, 69, 70, 76, 93, 40] can provide guaranteed robustness verification results, but they are unable to deal with the size of modern neural networks. Statistical methods are proposed in [6, 7, 13, 54, 81, 83, 84, 51] to assess the local robustness of deep neural networks with a probably approximately correct (PAC) guarantee, namely the network satisfies a probabilistic robustness property with a certain level of confidence. This type of method can better address the limitations of traditional robustness verification methods. In this work, we conduct research in this direction to investigate the robustness of trajectory prediction.

Interpretation Analysis. Deep learning systems have led to significant advancements in many aspects of our lives. However, their black-box nature poses challenges for many applications. It is generally difficult to rely on a system that cannot provide explanations for its decisions. This has spurred a substantial amount of research on explainable AI methods [44, 79, 59, 16, 60, 34, 36, 68, 30, 3, 32, 97], which supplement network predictions with explanations that humans can understand. However, there is currently limited research focused on providing explanations for the trajectory prediction of different methods. In our study, we train a PAC model to offer an interpretable analysis of the original model.

3. Problem Formulation

In this section, we present the formal modeling of trajectory prediction models and the formal specification of robustness in such models.

3.1. Trajectory Prediction

Denote by $x^t \in \mathbb{R}^2$ the spatial coordinate of an agent at timestamp t, then a trajectory over T timestamps is a sequence of the coordinates represented by a matrix $X \in \mathbb{R}^{2\times T}$. Considering the current timestamp as $t = 0$, we mark the timestamps as $t = -T_\mathrm{p} + 1, -T_\mathrm{p} + 2, \ldots, 0$ for a past trajectory over T_p timestamps. Then, let $X_0 \in \mathbb{R}^{2\times T}$ be the past trajectory of the to-be-predicted agent and $X_1, X_2, \ldots, X_N$ be those of N neighbouring agents. For $t = 1, 2, \ldots, T_\mathrm{f}$, we use Y_f to denote the ground truth of the future trajectory of the to-be-predicted agent. The goal of trajectory prediction is to train a prediction model $g : (\mathbb{R}^{2\times T_\mathrm{p}})^{N+1} \to \mathbb{R}^{2\times T_\mathrm{f}}$, so that the predicted future trajectory $Y = g(X_0, X_1, \ldots, X_N)$ is as close to the ground-truth Y_f as possible.

In current trajectory prediction models, stochastic prediction techniques have been introduced to capture the multi-modality of future movements, so the output of such trajectory prediction models are not deterministic, but probabilistic. In this work, due to the random mechanism widely used in trajectory prediction models, we consider the output $g(X_0, \ldots, X_N)$ of a trajectory prediction model as a probability distribution on the Borel measurable space $\mathbb{R}^{2\times T_\mathrm{f}}$. We write $Y \in g(X_0, \ldots, X_N)$ if Y is in the support of the probability distribution $g(X_0, \ldots, X_N)$.

3.2. Robustness of Prediction Models

Although existing works have explored the robustness of trajectory prediction models [95, 11, 37, 12, 98, 74], they fail to provide a formal definition of robustness. Instead, these works quantify robustness by their vulnerability to adversarial attacks. Therefore, we first provide rigorous definitions for robustness in the context of trajectory forecasting.

To describe a robustness region, we employ the L^∞-norm, which is most often used in robustness verification. For an input trajectory $\hat{X} \in \mathbb{R}^{2\times T}$, we consider any spatial coordinate x^t of the trajectory can be disturbed in the closed L^∞-norm ball with the center x^t and the radius $r > 0$. Then, we use $B(\hat{X}, r)$ to denote the set of the disturbing trajectories generated from $\hat{X}$, i.e., $B(\hat{X}, r) = \{X \in \mathbb{R}^{2\times T} \mid ||X - \hat{X}||_\infty \le r\}$.

Since trajectory prediction models are regression models, we cannot define local robustness as that in classification tasks, where the robustness property can be naturally given with the output scores. To define robustness in trajectory prediction models, we adapt the same intuition as that in global robustness [64], which requires that the output perturbation should be uniformly bounded. The output perturbation can be formalised as a metric $D : \mathbb{R}^{2\times T_\mathrm{f}} \times \mathbb{R}^{2\times T_\mathrm{f}} \to [0, +\infty)$. If we use the ground truth Y_f to measure the output perturbation, we have the following definition of label robustness:

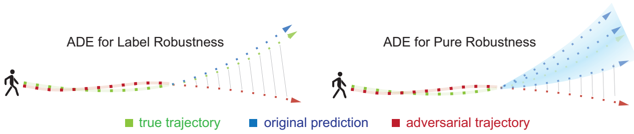

Figure 2. ADE of the adversarial trajectory prediction for label robustness and pure robustness.

Definition 1 (Label Robustness) *Let $\hat{\mathbf{X}} = (\hat{X}_0, \hat{X}_1, \ldots, \hat{X}_N)$ be the past trajectories of the to-be-predicted agent and its N neighbouring agents, and Y_{f} is ground truth of the future trajectories of the to-be-predicted agent. Given a prediction model g, an evaluation metric D, a safety constant s, then g is label-robust at $\hat{\mathbf{X}}$ w.r.t. the perturbation radius $r > 0$ if for any $X_i \in B(\hat{X}_i, r)$ $(i = 0, 1, \ldots, N)$ and any $Y \in g(X_0, X_1, \ldots, X_N)$, we have $D(Y, Y_{\mathrm{f}}) \leq s$.*

Robustness in models with random mechanism is quite different, where we require that $D(Y, Y_{\mathrm{f}}) \leq s$ for any possible output trajectory Y. In Def. 1, we always assume that the input $\mathbf{X}$ is chosen from the dataset, so that its ground truth Y_{f} is accessible. Since we measure the distance from the ground truth, a label-robust model intuitively has good performance in prediction and tolerance to adversaries. However, label robustness has the limitation that we must have the ground truth Y_{f}, so it is difficult to adapt it to the robustness regions where we do not know the ground truth. For such a consideration, we define pure robustness, where distance is measured from the output of $\hat{\mathbf{X}}$ in the model:

Definition 2 (Pure Robustness) *Let $\hat{\mathbf{X}} = (\hat{X}_0, \hat{X}_1, \ldots, \hat{X}_N)$ be the past trajectories of the to-be-predicted agent and its N neighbouring agents. Given a prediction model g, an evaluation metric D, a safety constant s, then g is purely robust at $\hat{\mathbf{X}}$ w.r.t. the perturbation radius $r > 0$ if for any $X_i \in B(\hat{X}_i, r)$ $(i = 0, 1, \ldots, N)$ and any $Y \in g(X_0, X_1 \ldots, X_N)$, there exists $\hat{Y} \in g(\hat{\mathbf{X}})$, s.t. $D(Y, \hat{Y}) \leq s$.*

We call it pure robustness since the distance is measured from the output of the model, in which situation only tolerance to adversaries is described. In Def. 2, we make more modifications for the random mechanism, since the output $g(\mathbf{X})$ is also a distribution. For an output trajectory Y, we look for a trajectory $\hat{Y} \in g(\hat{\mathbf{X}})$ such that their distance attains the minimum, and pure robustness requires that this minimum distance should be smaller than the safety constant s. In Fig. 2 we show the difference between label robustness and pure robustness.

To specify the definition of robustness, we still need to determine the evaluation metric D to measure the difference of two trajectories. Here we employ Average Displacement Error (ADE) [2, 1, 27, 48], which refers to the mean L^2 distance between all coordinates of ground truth and those of the predicted trajectory. For two trajectories Y_1 and Y_2 over timestamps $t = 1, 2, \ldots, T$, we generalise ADE with L^2 norm to measure the distance between them:

$$\mathrm{ADE}(Y_1, Y_2) = \frac{1}{T} \sum_{t=1}^{T} \|y_1^t - y_2^t\|_2.$$

In this work, we consider the label/pure robustness with $D = \mathrm{ADE}$. Note that other semantic metrics, such as metrics based on specific directions in [95], are also fully applicable to the above framework. In this work, we focus on robustness verification of trajectory prediction models:

> Given a trajectory prediction model g, we determine whether g is label-robust (or purely robust) at a given input $\hat{\mathbf{X}}$ w.r.t a given radius r.

4. Methodology

The most popular trajectory prediction models, including [21, 23, 72, 18, 52, 66, 94, 19, 55, 73, 26, 90], are all very large with stochastic output. As such, it is quite difficult to adopt traditional verification methods like SMT solving [40, 41] or abstract interpretation [70, 93] to verify their robustness properties.

In [51], Li et al. proposed a black-box DNN verification algorithm DeepPAC, where they relax the definition of robustness in a probabilistic way, allowing them to verify robustness at individual input regions using only a finite number of samples. This probably approximately correct (PAC) framework involves first learning a PAC model, an arbitrary function which (with probability close to 1 at a given confidence level) approximates the DNN at the input region within a margin of discrepancy. Next, using this PAC model we can verify the robustness at the input region with guarantees on the confidence and error rate. Due to its black-box nature, DeepPAC can be adapted to the robustness verification of trajectory prediction models. Moreover, the stochasticity of such models can be captured by PAC guarantees. We call our adapted method TrajPAC, and in this section we detail how TrajPAC is employed for robustness analysis of trajectory forecasting models.

4.1. PAC Model Learning

In Defs. 1 and 2, the robustness of a trajectory prediction model requires the distance between the perturbed prediction and the ground truth/original prediction to be bounded by a safety constant. Thus, to analyse the label/pure robustness, we learn a model approximating the corresponding distance $D(Y, \cdot)$ with the PAC guarantee and further infer its maximal values. Here we denote $\Delta(\mathbf{X})$ as the distribution $D(g(\mathbf{X}), \cdot)$, where $\mathbf{X} = (X_0^\top, \ldots, X_N^\top)$ and $X_i \in B(\hat{X}_i, r)$ for each i. Similar to DeepPAC, we choose the function template to be an affine function, i.e,

$\widetilde{\Delta}(\mathbf{X}) = \mathbf{X} \cdot \alpha + \beta$, where α and β are constant real vectors which will be learned from sampling. There are several reasons why we learn an affine function: First, the robustness properties we consider are all local robustness with a small neighboring region as the input region, and theoretically a continuous function can be approximated by an affine function with a very small error in a small region; after we learn the PAC model, we need to analyse how robust the PAC model is, and this analysis will be very easy and efficient if the PAC model is affine; also, an affine PAC model provides more accessible insight for model explanation.

To learn a function $\widetilde{\Delta}$ that fits Δ well, especially for a verification purpose, we desire that the difference of the two functions in the robustness region should be uniformly bounded by a margin $\lambda \geq 0$ as small as possible, so we have the following optimization problem:

$$\begin{aligned} & \min \lambda \\ \text{s.t.} \quad & \sup_{d \in \Delta(\mathbf{X})} |\widetilde{\Delta}(\mathbf{X}) - \Delta(\mathbf{X})| \leq \lambda, \\ & \forall \mathbf{X} \in B(X_0', r) \times \cdots \times B(X_N', r). \end{aligned} \tag{1}$$

Generally it is difficult to solve (1), since it has an uncountable number of constraints. Also, $\Delta(\mathbf{X})$ is stochastic in nature, which makes solving this optimisation problem nontrivial. Inspired by DeepPAC and [91], we can relax the problem (1) to finitely many constraints from the samples:

$$\begin{aligned} & \min \lambda \\ \text{s.t.} \quad & |\widetilde{\Delta}(\mathbf{X}) - \Delta(\mathbf{X})| \leq \lambda, \\ & \forall \mathbf{X} \in \mathcal{X}, \forall d \in \mathcal{D}(\mathbf{X}), \end{aligned} \tag{2}$$

where $\mathcal{X} \subseteq B(X_0', r) \times \cdots \times B(X_N', r)$ is a finite set of samples extracted independent and identically distributed from some distribution π, and $\mathcal{D}(\mathbf{X}) \subseteq D(g(\mathbf{X}), \cdot)$ is a finite set of samples from the distribution $D(g(\mathbf{X}), \cdot)$. This relaxation is slightly different from that in [51], because we need to sample not only in the robustness region, but also in the distribution of the output distance $D(g(\mathbf{X}), \cdot)$. The relaxed problem (2) is a linear programming (LP), whose optimal can be obtained efficiently. Since we only consider a finite subset of constraints, the optimal of (2) does not necessarily satisfy all the constraints in (1). In [51], a PAC guarantee can be constructed if we have enough samples, and we modify this result into our setting of trajectory prediction models, where stochastic output is considered.

Theorem 1 *Let $\epsilon > 0$ and $\eta > 0$ be the pre-defined error rate and the significance level, respectively, and K the number of samples. If*

$$K \geq \frac{2}{\epsilon}\left(\ln \frac{1}{\eta} + 2T_{\mathrm{p}}(N+1) + 1\right), \tag{3}$$

then with confidence at least $1 - \eta$, the optimal λ^* of (2) satisfies all the constraints in (1) but at most a fraction of probability ϵ, i.e., $\mathbb{P}\left(|\widetilde{\Delta}(\mathbf{X}) - \Delta(\mathbf{X})| \geq \lambda^*\right) \leq \epsilon$, where the probability measure $\mathbb{P}$ is the independent coupling of the sampling distribution π and the random mechanism in the model $g(\cdot)$.

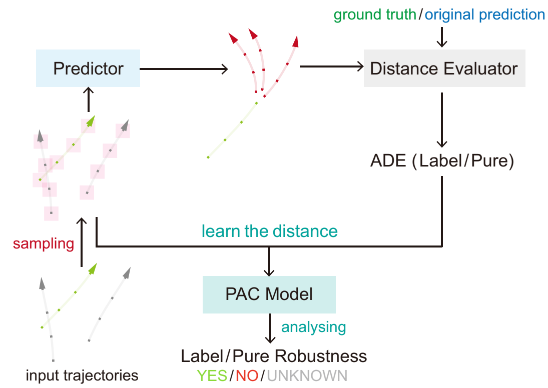

Figure 3. Framework of robustness analysis

Thm. 1 generalises the DeepPAC method to trajectory prediction models, where we are faced with a regression model with random output, and different robustness properties. The essential difference is that the probability distribution $\mathbb{P}$, which is used for describing the PAC guarantee, is not the sampling distribution, but its coupling with the random mechanism of the model. The proof of Thm. 1 can be found in Appendix A.

Now our black-box framework of robustness analysis for trajectory prediction models is explicit, as is shown in Fig. 3. Given the error rate ϵ and the significance level η, we extract K samples in $B(X_0, r) \times \cdots \times B(X_N, r)$, where K satisfies (3). With the samples, we construct the linear programming problem (2) and obtain (one of) its optimal, which gives the coefficients α and β in the PAC model $\widetilde{\Delta}$ and the margin λ^*, and they will further help us analyse how robust the model is.

The optimisation of focused learning proposed in [51] still fits in our settings, and we use it in our implementation. More details can be found in Appendix B.

4.2. Robustness Analysis

We follow a similar robustness analysis procedure to DeepPAC. When the optimisation problem (2) is solved, we obtain the PAC model $\widetilde{\Delta}$ as well as the optimal margin λ^*. Intuitively, $\widetilde{\Delta}(\mathbf{X}) \pm \lambda^*$ approximates the upper/lower bound of $\Delta(\mathbf{X})$ in the robustness region with the PAC guarantee. It is easy to see that, if the maximum of $\widetilde{\Delta}(\mathbf{X}) + \lambda^*$ is smaller than what the robustness property requires, i.e., the parameter s in Def. 1 or Def. 2, then it holds under the same PAC guarantee. Since $\widetilde{\Delta}$ is an affine function, its maximum in a box region can be easily computed.

There are three circumstances that may occur in the robustness analysis:

- The maximum of $\widetilde{\Delta}(\mathbf{X}) + \lambda^*$ is smaller than s. In this case, the robustness property holds with a PAC guarantee, and actually the models satisfies the so-call PAC-model robustness defined in [51]. The analysis outputs YES. It is worth mentioning that, PAC-model robustness is far stronger than PAC robustness, especially that obtained from statistical methods like hypothesis testing or confidence interval calculation, because we have a model $\widetilde{\Delta}$ witnessing the robustness property PAC-true.
- The maximum of $\widetilde{\Delta}(\mathbf{X}) + \lambda^*$ is strictly larger than s, and we can find a true counterexample. In the model learning phase, if there exists a sample that violates the property, then it is a true counterexample. Also, when we calculate the maximum of $\widetilde{\Delta}(\mathbf{X}) + \lambda^*$, the maximum point $\arg\max \widetilde{\Delta}(\mathbf{X})$ is likely to be a counterexample, and we run it in the original model to see whether it is a true counterexample. Once the PAC-model robustness does not hold, and we find a true counterexample in either way, the analysis outputs NO, i.e., the robustness property does not hold, with a true counterexample.
- It may occur that the maximum of $\widetilde{\Delta}(\mathbf{X})+\lambda^*$ is strictly larger than s, but we cannot find a true counterexample. In this case, it is not sufficient to judge whether the model is robust or not according to the learned PAC model, so the analysis outputs UNKNOWN.

We remark that, in the first circumstance where the PAC-model robustness holds, we do not further check whether there is a true counterexample, because even if it exists, it does not violate the PAC-model robustness, in which the violation of the robustness property may occur with probability no more than the error rate ϵ.

4.3. Interpretation Analysis

The PAC model we learn can also provide insight into two key features used by forecasting models when making predictions: the **critical paths** and the **critical steps** of agents. These two features are intuitive in the real world. For example, the movements of a person in front of you are more significant than the movements of someone behind you, and certain steps (e.g., changing direction) have greater impact than others.

As our PAC model is an affine function, there is a corresponding coefficient for every spatial coordinate in the input trajectory. The greater this coefficient magnitude is, the greater the impact of the corresponding coordinate's change on the prediction's label/pure ADE. We denote the l^∞ normalized vector of coefficient magnitudes as the *sensitivity* of our PAC model.

Therefore, spatial coordinates with high sensitivity values are identified as the critical steps, and trajectories with high average sensitivities are the critical paths. These critical steps and paths reflect which features in the historical trajectories can lead to vulnerabilities in the model. Not only does this give us a more interpretable understanding of how the model makes predictions, but it also allows us to analyse the key features that affect the model's robustness. Additionally, we can handcraft potential adversaries by only perturbing these key features, making our counterexamples highly intuitive.

5. Experiments

In this section, we evaluate our PAC-model robustness analysis method. We implement our algorithm TRAJPAC as a prototype. Its implementation is based on Python 3.7.8. Experiments are conducted on a Windows 11 PC with AMD R7, GTX 3070Ti, and 16G RAM. All the implementation and data used in this section are publicly available[1].

Datasets. We evaluate our method using the public pedestrian trajectories forecasting benchmarks ETH/UCY [61, 46] and the Stanford Drone Dataset (SDD) [63]. The ETH and UCY dataset group consists of five different scenes – ETH and HOTEL (from ETH), and UNIV, ZARA1 and ZARA2 (from UCY) and all the scenes report the position of pedestrians in world coordinates and hence the results are in metres. All the prediction models in our paper use the "leave one-out" method [27, 33, 43, 66] for training and evaluation. We follow the existing works that observing 8 frames (3.2 seconds) trajectories and predicting the next 12 frames (4.8 seconds). We randomly choose three predicted trajectories from each scenes for analysis, noted as (frame ID , person ID). Experiments regarding SDD can be found in Appendix D.

Prediction Models. In our paper, we analyse four state-of-art multi-model prediction models: Trajectron++ [66], AgentFormer [94], MemoNet [90] and MID [26].

Sampling. The sampling distribution π is the uniform distribution on the robustness region. When we calculate the ADE of the samples, we use a modified version of ADE, the minimum average displacement error of K trajectory samples, which is a standard metric for trajectory prediction [27, 65, 66, 62, 17]. We claim that this will not break the PAC guarantee in Thm. 1. More details can be found in Appendix C. In our experiment, we choose $K = 20$.

Implementation details. In the later part, we choose 1 meter/0.5 meters to be the safety constant for label/pure robustness analysis with the perturbation radius $r = 0.03$ meters, respectively. Experiments with varying values of r can be found in Appendix E. As for PAC model learning, we choose $\eta = 0.01$ and $\epsilon = 0.01$.

In what follows, we are going to answer the research questions below:

[1] https://github.com/ZL-Helios/TrajPAC

Scene	ID	Label Robustness				Pure Robustness			
		Traj++	Memo	AgentF	MID	Traj++	Memo	AgentF	MID
ETH	(4400, 79)	○	✓	✗	✗	○	✓	✗	✓
	(6490, 127)	✓	✓	✗	✗	✓	✓	✗	✓
	(10340, 257)	○	○	✗†	✗	○	✓	✗	✗
Hotel	(7550, 157)	✓	✓	✗	○	✓	✓	✓	✓
	(10530, 236)	✓	✓	○	✓	✓	✓	✓	✓
	(15030, 345)	✓	✓	✓	✓	✓	✓	✓	✓
Zara1	(4430, 69)	○	✓	✗	✗	✓	✓	✓	✓
	(6050, 102)	✓	✓	✗	✗	✓	✓	○	✓
	(8680, 142)	✗†	○	✗†	✓	○	✓	✓	✓
Zara2	(3400, 65)	✓	✓	✗	✓	✓	✓	✓	✓
	(7430, 141)	✓	✓	✗	✗	✓	✓	○	○
	(10030, 195)	✗	✓	✗	✗	✗	○	○	✓
Univ	(1840, 105)	✗	✗	✗†	✗	○	✓	○	✓
	(4820, 202)	✗†	✗	✗†	○	○	✓	○	✓
	(5250, 297)	✓	✓	✗	✓	○	✓	○	✓

Table 1. Label/pure robustness verification. We mark ✓ if it is PAC-model robust, i.e., the robustness analysis returns YES, ✗ if the PAC-model with the optimal margin is not robust and we find a true counterexample, i.e., the robustness analysis returns NO, and ○ otherwise, i.e., the robustness analysis returns UNKNOWN. We use † to indicate that PGD attacks successfully, i.e., the adversary found by PGD exceeds the robustness threshold.

Method	Average Sampling Rate (iteration/s)					Average PAC-Model Learning Time (s)
	ETH	Hotel	Zara1	Zara2	Univ	
Traj++	51.50	52.15	52.27	51.87	52.26	1.02
MemoNet	0.99	1.92	1.91	1.32	0.94	1.05
AgentFormer	15.57	16.22	16.30	15.38	12.38	1.10
MID	0.14	0.13	0.13	0.14	0.13	0.15

Table 2. The average sampling rate, in iterations per second, of each model at each scene. The diffusion-based model (MID) has the longest sampling rate, in which 10000 samples require a time of ∼20 hours. Because of this, we opt to use fewer samples for its scenario optimization process, resulting in the faster PAC learning time.

RQ1: Does TRAJPAC perform well in verifying robustness?
RQ2: Can TRAJPAC precisely capture the robustness performance of the prediction models?
RQ3: Can TRAJPAC provide intuitive analysis of the robustness performance of different prediction models?

5.1. Robustness Analysis of Different Models

First, we evaluate the performance of TRAJPAC on giving robustness verification. This includes whether the model can achieve robustness prediction for a given safety constant and perturbation radius, whether it can be applied to a wide range of models with high validation efficiency, and whether those cases verified as robust demonstrate good anti-attack performance.

As shown in Tab. 1, as a black-box method, TRAJPAC can analyse label and pure robustness of different trajectory prediction models, showing good scalability. The sampling time varies among different prediction models though, yet time for PAC-model learning and robustness analysis is quite short, as shown in Tab. 2. This demonstrates that TRAJPAC is very efficient in analysing large trajectory prediction models.

TRAJPAC only provides a PAC guarantee, so we are concerned with the soundness of its robustness analysis. We conduct PGD attacks on Trajectron++ and AgentFormer like [11]. For the cases TRAJPAC outputs YES, PGD does not find any true counterexamples, which implies that TRAJPAC is sound empirically. TRAJPAC is conservative in analysing robustness: Even on the UNKNOWN cases, there is no successful PGD attack. Also, TRAJPAC shows good performance in finding counterexamples, as TRAJPAC can find counterexamples to which PGD does not get access.

Scene	ADE_{20} (in metre), best-of-20 samples			
	Traj++	MemoNet	AgentFormer	MID
ETH	0.46	0.41	0.41	0.51
Hotel	0.15	0.14	0.30	0.15
Zara1	0.49	0.57	0.34	0.25
Zara2	0.36	0.33	0.27	0.27
Univ	0.69	0.54	0.62	0.31
Average ADE	0.43	0.40	0.39	0.30

Table 3. Average predicted ADE_{20} scores for the three verification samples per scene.

Due to the model performance on motion forecasting affecting its label robustness, we provide the predicted ADE in Tab. 3. The four methods show relatively similar average performance without perturbation, so in terms of maintaining accurate predictions in the face of perturbation, Memonet does exhibit stronger label robustness, as emphasised in [90]. Similar to the findings in [11], Trajectron++ exhibits stronger label robustness compared to Agentformer. We can see that the analysis results given by TRAJPAC are consistent with other methods.

> **Answer RQ1:** TRAJPAC shows good scalability, efficiency and soundness in robustness analysis of different trajectory prediction models. Its results of robustness analysis are consistent with other methods.

5.2. Precision of the PAC models

Generally it is difficult to straight evaluate how precise the PAC models learned by TRAJPAC are. Here we calculate four ADE estimations, namely the ADE upper bound given by the PAC model, the ADE of the adversary generated by our PAC model, the maximum ADE among the samples required for training TRAJPAC, and the ADE of the adversary from PGD attack; the first two are estimations from the PAC model, while the latter two are ADE performance of the model in the robustness region. In Fig. 4, we present a detailed illustration of the four estimations.

First, the ADE upper bound given by TRAJPAC is very close to (and still above) the maximum sampled ADE during the model learning process, indicating that our PAC model has captured the behavior of the original model well and produced highly accurate ADE upper bounds, as shown in Tab. 4. Furthermore, the robustness analysis results obtained through our method still exhibit significant soundness in Fig. 4, as the ADE of the linear adversary generated from PAC model, as well as PGD adversary, are all smaller than the ADE upper bound. Also, we notice that the adversaries generated by our PAC model are as effective as

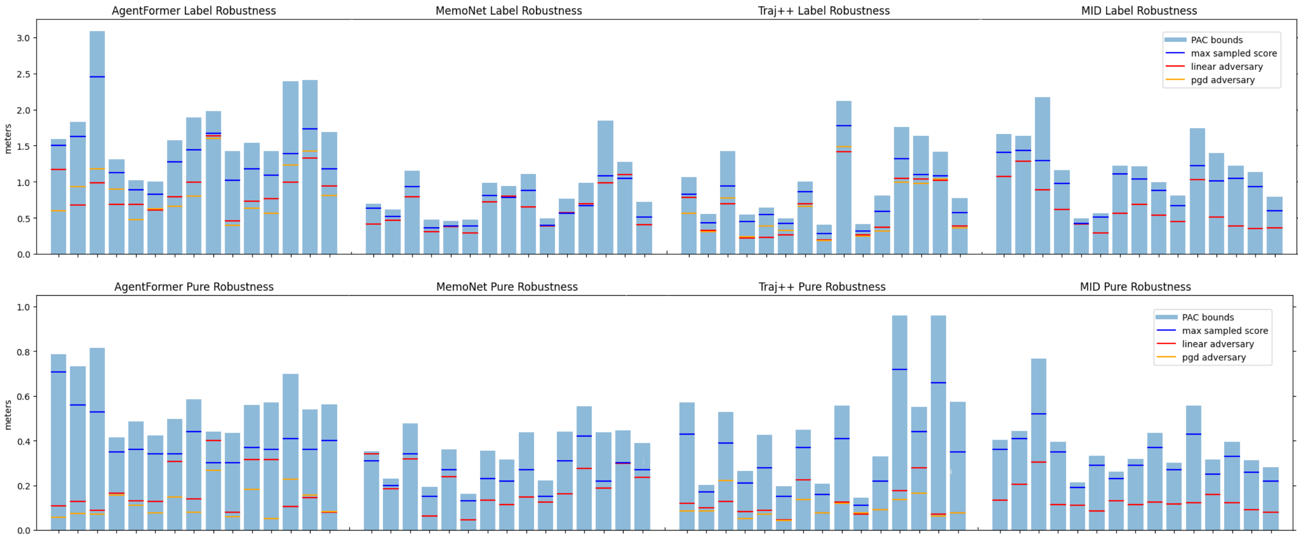

Figure 4. Visualizations of the PAC ADE upper bounds (blue bars), maximum sampled ADE encountered in the PAC model learning process (blue stripes), and ADE of linear and pgd adversaries (red and orange stripes respectively) from our PAC model and PGD attacks.

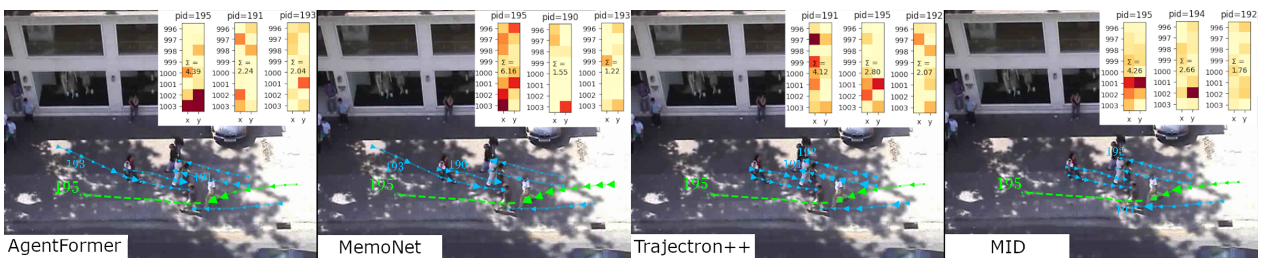

Figure 5. Sensitivity plots for each prediction model at sample (10030, 195) from scene Zara2. The green path is the agent trajectory and the blue paths are neighboring trajectories. The size of the directional arrows are proportional to the sensitivity of our PAC linear model at that position. The top right of each plot contains a heatmap of the top three critical paths. Darker colors in the heatmap represent higher sensitivity values. The value inside each heatmap is the sum of all sensitivities in the path.

Scene	PAC ADE upper bound - max sampled ADE (label robustness / pure robustness)			
	Traj++	MemoNet	AgentFormer	MID
ETH	0.28 / 0.10	0.13 / 0.07	0.31 / 0.18	0.45 / 0.11
Hotel	0.08 / 0.08	0.09 / 0.06	0.16 / 0.09	0.10 / 0.04
Zara1	0.21 / 0.09	0.19 / 0.13	0.35 / 0.15	0.13 / 0.04
Zara2	0.25 / 0.13	0.20 / 0.11	0.37 / 0.18	0.35 / 0.07
Univ	0.36 / 0.21	0.40 / 0.16	0.73 / 0.21	0.19 / 0.06
Average	0.24 / 0.12	0.20 / 0.11	0.38 / 0.16	0.24 / 0.06

Table 4. Differences between our computed PAC ADE upper bound and the maximum sampled ADE during the model learning process.

Scene	$\lvert \mathrm{ADE}_{\mathrm{pgd}} - \mathrm{ADE}_{\mathrm{linear}} \rvert$ (label robustness / pure robustness)	
	Traj++	AgentFormer
ETH	0.10 / 0.05	0.34 / 0.04
Hotel	0.08 / 0.02	0.15 / 0.03
Zara1	0.04 / 0.03	0.12 / 0.12
Zara2	0.05 / 0.02	0.12 / 0.14
Univ	0.04 / 0.04	0.16 / 0.05
Average	0.06 / 0.03	0.18 / 0.08

Table 5. ADE of adversaries from PAC models and PGD.

PGD, since the ADE of the adversary generated by our PAC model is very close to that of PGD adversary, as is depicted in Tab. 5. From Fig. 4, the adversaries generated by our method exhibit better overall attack effectiveness compared to PGD in the analysis of pure robustness.

Answer RQ2: TrajPAC can provide tight ADE upper bound of different prediction methods. Adversaries generated from TrajPAC exhibit comparable (and even better) performance to adversaries found by PGD.

5.3. Interpretation Analysis

We perform an interpretation analysis on the sample (10030, 195) in Zara2. Among the four methods, MemoNet is the only label-robust method, with a label ADE upper bound of 0.98. Trajectron++ is the least robust, with an upper bound of 1.76. In Fig. 5 we visualise the critical steps of different prediction methods, and shows the top three critical paths in each method. Based on Fig. 5, we emphasise the following observations: Steps closer to the present are more likely to be critical steps, and the trajectory of the agent it-self is often the critical path.

Our analysis also exposes potential vulnerabilities at each sample. For instance, the critical paths captured by MemoNet (190 and 193) are walking directly towards the agent, whereas the critical paths captured by Trajectron++ (191 and 192) are walking away. Knowing this, black-box attackers are able to handcraft adversaries by adding perturbations to only these key positions. In particular, the critical paths of Trajectron++ makes it more susceptible to attacks, since defenses are more likely to focus on the paths walking directly towards the agent, rather than those walking away.

Answer RQ3: TRAJPAC can identify key features that contribute to the overall performance and robustness through sensitivity analysis of the PAC model.

6. Conclusion

We present TRAJPAC for robustness verification of trajectory prediction models. It is highly scalable, efficient, empirically sound, and capable of generating adversaries and interpretation. As for future works, we will consider more realistic safety properties in trajectory prediction, and how to use the verification results of trajectory prediction to analyse the safety of autonomous driving scenarios.

Acknowledgements

Supported by CAS Project for Young Scientists in Basic Research, Grant No.YSBR-040, and ISCAS New Cultivation Project ISCAS-PYFX-202201.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016. 2, 4

[2] Alexandre Alahi, Vignesh Ramanathan, and Li Fei-Fei. Socially-aware large-scale crowd forecasting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2203–2210, 2014. 4

[3] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018. 3

[4] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[5] Inhwan Bae, Jin-Hwi Park, and Hae-Gon Jeon. Non-probability sampling network for stochastic human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6477–6487, 2022. 1

[6] Teodora Baluta, Zheng Leong Chua, Kuldeep S Meel, and Prateek Saxena. Scalable quantitative verification for deep neural networks. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 312–323. IEEE, 2021. 3

[7] Teodora Baluta, Shiqi Shen, Shweta Shinde, Kuldeep S Meel, and Prateek Saxena. Quantitative verification of neural networks and its security applications. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1249–1264, 2019. 3

[8] Akhilan Boopathy, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel. Cnn-cert: An efficient framework for certifying robustness of convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3240–3247, 2019. 3

[9] Giuseppe Carlo Calafiore and Marco C. Campi. The scenario approach to robust control design. *IEEE Trans. Autom. Control.*, 51(5):742–753, 2006. 14

[10] Marco C. Campi, Simone Garatti, and Maria Prandini. The scenario approach for systems and control design. *Annu. Rev. Control.*, 33(2):149–157, 2009. 14

[11] Yulong Cao, Chaowei Xiao, Anima Anandkumar, Danfei Xu, and Marco Pavone. Advdo: Realistic adversarial attacks for trajectory prediction. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*, pages 36–52. Springer, 2022. 1, 3, 7

[12] Yulong Cao, Danfei Xu, Xinshuo Weng, Zhuoqing Mao, Anima Anandkumar, Chaowei Xiao, and Marco Pavone. Robust trajectory prediction against adversarial attacks. *arXiv preprint arXiv:2208.00094*, 2022. 1, 3

[13] Luca Cardelli, Marta Kwiatkowska, Luca Laurenti, Nicola Paoletti, Andrea Patane, and Matthew Wicker. Statistical guarantees for the robustness of bayesian neural networks. *arXiv preprint arXiv:1903.01980*, 2019. 3

[14] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019. 3

[15] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017. 3

[16] Brandon Carter, Siddhartha Jain, Jonas W Mueller, and David Gifford. Overinterpretation reveals image classification model pathologies. *Advances in Neural Information Processing Systems*, 34:15395–15407, 2021. 3

[17] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019. 6, 15

[18] Guangyi Chen, Junlong Li, Nuoxing Zhou, Liangliang Ren, and Jiwen Lu. Personalized trajectory prediction via distribution discrimination. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15580–15589, 2021. 2, 4

[19] Hao Cheng, Wentong Liao, Xuejiao Tang, Michael Ying Yang, Monika Sester, and Bodo Rosenhahn. Exploring dynamic context for multi-path trajectory prediction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12795–12801. IEEE, 2021. 2, 4

[20] Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In *28th USENIX security symposium (USENIX security 19)*, pages 321–338, 2019. 3

[21] Patrick Dendorfer, Sven Elflein, and Laura Leal-Taixé. Mg-gan: A multi-generator model preventing out-of-distribution samples in pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13158–13167, 2021. 2, 4

[22] Jinghai Duan, Le Wang, Chengjiang Long, Sanping Zhou, Fang Zheng, Liushuai Shi, and Gang Hua. Complementary attention gated network for pedestrian trajectory prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 542–550, 2022. 1

[23] Liangji Fang, Qinhong Jiang, Jianping Shi, and Bolei Zhou. Tpnet: Trajectory proposal network for motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6797–6806, 2020. 2, 4

[24] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Gd-gan: Generative adversarial networks for trajectory prediction and group detection in crowds. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part I 14*, pages 314–330. Springer, 2019. 2

[25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[26] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17113–17122, 2022. 2, 3, 4, 6

[27] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2255–2264, 2018. 2, 4, 6, 15

[28] Abdullah Hamdi, Sara Rojas, Ali Thabet, and Bernard Ghanem. Advpc: Transferable adversarial perturbations on 3d point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 241–257. Springer, 2020. 3

[29] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995. 2

[30] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 3–19. Springer, 2016. 3

[31] Lifeng Huang, Chengying Gao, Yuyin Zhou, Cihang Xie, Alan L Yuille, Changqing Zou, and Ning Liu. Universal physical camouflage attacks on object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 720–729, 2020. 3

[32] Wei Huang, Xingyu Zhao, Gaojie Jin, and Xiaowei Huang. Safari: Versatile and efficient evaluations for robustness of interpretability. *ICCV*, 2023. 3

[33] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6272–6281, 2019. 6

[34] Aya Abdelsalam Ismail, Hector Corrada Bravo, and Soheil Feizi. Improving deep learning interpretability by saliency guided training. *Advances in Neural Information Processing Systems*, 34:26726–26739, 2021. 3

[35] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2375–2384, 2019. 2

[36] Jeya Vikranth Jeyakumar, Joseph Noor, Yu-Hsi Cheng, Luis Garcia, and Mani Srivastava. How can i explain this to you? an empirical study of deep neural network explanation methods. *Advances in Neural Information Processing Systems*, 33:4211–4222, 2020. 3

[37] Ruochen Jiao, Xiangguo Liu, Takami Sato, Qi Alfred Chen, and Qi Zhu. Semi-supervised semantics-guided adversarial training for trajectory prediction. *arXiv preprint arXiv:2205.14230*, 2022. 1, 3

[38] Gaojie Jin, Xinping Yi, Wei Huang, Sven Schewe, and Xiaowei Huang. Enhancing adversarial training with second-order statistics of weights. In *CVPR*, 2022. 3

[39] Gaojie Jin, Xinping Yi, Dengyu Wu, Ronghui Mu, and Xiaowei Huang. Randomized adversarial training via taylor expansion. In *CVPR*, 2023. 3

[40] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*, pages 97–117. Springer, 2017. 3, 4

[41] Guy Katz, Derek A Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah, Shantanu Thakoor, Haoze Wu, Aleksandar Zeljić, et al. The marabou framework for verification and analysis of deep neural networks. In *International Conference on Computer Aided Verification*, pages 443–452. Springer, 2019. 3, 4

[42] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV 12*, pages 201–214. Springer, 2012. 2

[43] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. *Advances in Neural Information Processing Systems*, 32, 2019. 2, 6

[44] Peter Cho-Ho Lam, Lingyang Chu, Maxim Torgonskiy, Jian Pei, Yong Zhang, and Lanjun Wang. Finding representative interpretations on convolutional neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1345–1354, 2021. 3

[45] 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. 2

[46] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007. 6

[47] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, et al. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE intelligent vehicles symposium (IV)*, pages 163–168. IEEE, 2011. 1

[48] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. Conditional generative neural system for probabilistic trajectory prediction. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6150–6156. IEEE, 2019. 2, 4

[49] Lihuan Li, Maurice Pagnucco, and Yang Song. Graph-based spatial transformer with memory replay for multi-future pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2231–2241, 2022. 1

[50] Renjue Li, Jianlin Li, Cheng-Chao Huang, Pengfei Yang, Xiaowei Huang, Lijun Zhang, Bai Xue, and Holger Hermanns. Prodeep: a platform for robustness verification of deep neural networks. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1630–1634, 2020. 3

[51] Renjue Li, Pengfei Yang, Cheng-Chao Huang, Youcheng Sun, Bai Xue, and Lijun Zhang. Towards practical robustness analysis for dnns based on pac-model learning. In *Proceedings of the 44th International Conference on Software Engineering*, pages 2189–2201, 2022. 2, 3, 4, 5, 6, 14, 15

[52] Yuejiang Liu, Qi Yan, and Alexandre Alahi. Social nce: Contrastive learning of socially-aware motion representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15118–15129, 2021. 2, 4

[53] 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. 2

[54] Ravi Mangal, Aditya V Nori, and Alessandro Orso. Robustness of neural networks: A probabilistic and practical approach. In *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, pages 93–96. IEEE, 2019. 3

[55] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 759–776. Springer, 2020. 1, 2, 4

[56] Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *2009 IEEE conference on computer vision and pattern recognition*, pages 935–942. IEEE, 2009. 2

[57] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14424–14432, 2020. 1

[58] Jeremy Morton, Tim A Wheeler, and Mykel J Kochenderfer. Analysis of recurrent neural networks for probabilistic modeling of driver behavior. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1289–1298, 2016. 2

[59] Meike Nauta, Ron Van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14933–14943, 2021. 3

[60] Jayneel Parekh, Pavlo Mozharovskyi, and Florence d'Alché Buc. A framework to learn with interpretation. *Advances in Neural Information Processing Systems*, 34:24273–24285, 2021. 3

[61] Stefano Pellegrini, Andreas Ess, and Luc Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*, pages 452–465. Springer, 2010. 6

[62] Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14074–14083, 2020. 6, 15

[63] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 549–565. Springer, 2016. 6

[64] Wenjie Ruan, Min Wu, Youcheng Sun, Xiaowei Huang, Daniel Kroening, and Marta Kwiatkowska. Global robustness evaluation of deep neural networks with provable guarantees for the hamming distance. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5944–5952, 2019. 3

[65] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and

physical constraints. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1349–1358, 2019. 2, 6, 15

[66] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 683–700. Springer, 2020. 2, 3, 4, 6, 15

[67] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Mo Zhou, Zhenxing Niu, and Gang Hua. Sgcn: Sparse graph convolution network for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8994–9003, 2021. 1

[68] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017. 3

[69] Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. *Advances in neural information processing systems*, 31, 2018. 3

[70] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–30, 2019. 3, 4

[71] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015. 2

[72] Hao Sun, Zhiqun Zhao, and Zhihai He. Reciprocal learning networks for human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7416–7425, 2020. 2, 4

[73] Jianhua Sun, Yuxuan Li, Hao-Shu Fang, and Cewu Lu. Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13250–13259, 2021. 2, 4

[74] Kaiyuan Tan, Jun Wang, and Yiannis Kantaros. Targeted adversarial attacks against neural network trajectory predictors. *arXiv preprint arXiv:2212.04138*, 2022. 1, 3

[75] Charlie Tang and Russ R Salakhutdinov. Multiple futures prediction. *Advances in neural information processing systems*, 32, 2019. 2

[76] Hoang-Dung Tran, Xiaodong Yang, Diego Manzanas Lopez, Patrick Musau, Luan Viet Nguyen, Weiming Xiang, Stanley Bak, and Taylor T Johnson. Nnv: the neural network verification tool for deep neural networks and learning-enabled cyber-physical systems. In *International Conference on Computer Aided Verification*, pages 3–17. Springer, 2020. 3

[77] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[78] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In *2018 IEEE international Conference on Robotics and Automation (ICRA)*, pages 4601–4607. IEEE, 2018. 2

[79] Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. Interpretable image recognition by constructing transparent embedding space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 895–904, 2021. 3

[80] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):283–298, 2007. 2

[81] Stefan Webb, Tom Rainforth, Yee Whye Teh, and M Pawan Kumar. A statistical approach to assessing neural network robustness. *arXiv preprint arXiv:1811.07209*, 2018. 3

[82] Yuxin Wen, Jiehong Lin, Ke Chen, CL Philip Chen, and Kui Jia. Geometry-aware generation of adversarial point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):2984–2999, 2020. 3

[83] Lily Weng, Pin-Yu Chen, Lam Nguyen, Mark Squillante, Akhilan Boopathy, Ivan Oseledets, and Luca Daniel. Proven: Verifying robustness of neural networks with a probabilistic approach. In *International Conference on Machine Learning*, pages 6727–6736. PMLR, 2019. 3

[84] Matthew Wicker, Luca Laurenti, Andrea Patane, and Marta Kwiatkowska. Probabilistic safety for bayesian neural networks. In *Conference on Uncertainty in Artificial Intelligence*, pages 1198–1207. PMLR, 2020. 3

[85] Chong Xiang, Charles R Qi, and Bo Li. Generating 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9136–9144, 2019. 3

[86] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018. 3

[87] Chaowei Xiao, Dawei Yang, Bo Li, Jia Deng, and Mingyan Liu. Meshadv: Adversarial meshes for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6898–6907, 2019. 3

[88] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1369–1378, 2017. 3

[89] Chenxin Xu, Maosen Li, Zhenyang Ni, Ya Zhang, and Siheng Chen. Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6507, 2022. 1

[90] Chenxin Xu, Weibo Mao, Wenjun Zhang, and Siheng Chen. Remember intentions: retrospective-memory-based trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6488–6497, 2022. 3, 4, 6, 7

[91] Bai Xue, Miaomiao Zhang, Arvind Easwaran, and Qin Li. Pac model checking of black-box continuous-time dynamical systems. *IEEE Transactions on Computer-Aided Design*

of Integrated Circuits and Systems, 39(11):3944–3955, 2020. 5

[92] Chenglin Yang, Adam Kortylewski, Cihang Xie, Yinzhi Cao, and Alan Yuille. Patchattack: A black-box texture-based attack with reinforcement learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI*, pages 681–698. Springer, 2020. 3

[93] Pengfei Yang, Renjue Li, Jianlin Li, Cheng-Chao Huang, Jingyi Wang, Jun Sun, Bai Xue, and Lijun Zhang. Improving neural network verification through spurious region guided refinement. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 389–408. Springer, 2021. 3, 4

[94] 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. 2, 3, 4, 6

[95] Qingzhao Zhang, Shengtuo Hu, Jiachen Sun, Qi Alfred Chen, and Z Morley Mao. On adversarial robustness of trajectory prediction for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15159–15168, 2022. 1, 3, 4

[96] 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. 2

[97] Xingyu Zhao, Wei Huang, Xiaowei Huang, Valentin Robu, and David Flynn. Baylime: Bayesian local interpretable model-agnostic explanations. In *UAI*. PMLR, 2021. 3

[98] Zhihao Zheng, Xiaowen Ying, Zhen Yao, and Mooi Choo Chuah. Robustness of trajectory prediction models under map-based attacks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4541–4550, 2023. 1, 3