

Are socially-aware trajectory prediction models really socially-aware?

Saeed Saadatnejad ^{a,1,*}, Mohammadhossein Bahari ^{a,1,*}, Pedram Khorsandi ^{b,2},
 Mohammad Saneian ^{b,2}, Seyed-Mohsen Moosavi-Dezfooli ^c, Alexandre Alahi ^a

^a EPFL, Switzerland

^b Sharif University of Technology, Iran

^c Imperial College London, United Kingdom

ARTICLE INFO

Keywords:

Human trajectory prediction
 Human social behavior simulation
 Adversarial attack
 Robustness

ABSTRACT

Our transportation field has recently witnessed an arms race of neural network-based trajectory predictors. While these predictors are at the core of many applications such as autonomous navigation or pedestrian flow simulations, their adversarial robustness has not been carefully studied. In this paper, we introduce a socially-attended attack to assess the social understanding of prediction models in terms of collision avoidance. An attack is a small yet carefully-crafted perturbations to fail predictors. Technically, we define collision as a failure mode of the output, and propose hard- and soft-attention mechanisms to guide our attack. Thanks to our attack, we shed light on the limitations of the current models in terms of their social understanding. We demonstrate the strengths of our method on the recent trajectory prediction models. Finally, we show that our attack can be employed to increase the social understanding of state-of-the-art models. The code is available at <https://s-attack.github.io/>.

1. Introduction

Understanding the social behavior of humans is a core problem in many transportation applications ranging from autonomous navigation (e.g., social/delivery robots (Bennewitz et al., 2002) or autonomous vehicles (Bouhsain et al., 2020; Bahari et al., 2021)), to microscopic pedestrian flow simulations (Seer et al., 2014; Robin et al., 2009). For a robot to navigate among crowds safely or for an autonomous vehicle to drive in urban areas harmlessly, human behavior anticipation is essential. In particular, dealing with humans makes the problem safety-critical. For instance, a self-driving car's wrong prediction in a crosswalk can put a pedestrian's life in danger. Being a safety-critical problem raises the need for careful assessments of the trajectory prediction methods to mitigate the risks associated with humans. Consequently, the robustness properties of those methods, as one of the important assessment aspects, should be carefully studied.

The pedestrian trajectory prediction problem is to predict future positions of pedestrians given their past positions as inputs. The problem has received solutions based on microscopic human simulation models (Blue and Adler, 2001; Duives et al., 2013). Recently, diverse approaches based on neural networks are proposed. Various models based on Long-Short-Term-Memory networks (Alahi et al., 2016), convolutional neural networks (Nikhil and Tran Morris, 2018), and Generative Adversarial Networks (GAN) (Gupta

* Corresponding author.

E-mail addresses: saeed.saadatnejad@epfl.ch (S. Saadatnejad), mohammadhossein.bahari@epfl.ch (M. Bahari).

¹ Equal contribution as the first authors.

² Equal contribution as the second authors.

<https://doi.org/10.1016/j.trc.2022.103705>

Received 2 September 2021; Received in revised form 11 February 2022; Accepted 25 April 2022

Available online 13 June 2022

0968-090X/© 2022 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

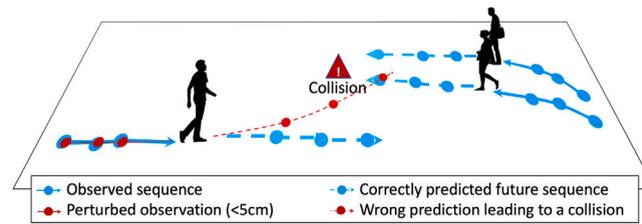


Fig. 1. Given the observation trajectories of the agents in the scene, a predictor (here S-LSTM (Alahi et al., 2016)) forecasts the future positions reasonably (blue lines). However, with less than 5 cm perturbation in the observation trajectory (in red), an unacceptable collision is predicted. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

et al., 2018) are presented. The main challenge of the problem lies in learning the interactions between people. Therefore, socially-aware neural network-based models are designed to tackle the interaction more accurately (Alahi et al., 2016; Vemula et al., 2018; Kosaraju et al., 2019; Mohamed et al., 2020). Human interactions involve different social behaviors such as collision avoidance, walking within a group, and merging from different directions into a specific point. Among all behaviors, collision avoidance, *i.e.*, people choosing a path that avoids collision with others, is one of the key behaviors rarely violated. That is why many previous works consider respecting collision avoidance as the evidence of their model being social (Mohamed et al., 2020; Kosaraju et al., 2019; Kothari et al., 2021). Similarly, we consider collision avoidance as an indicator of social behavior of the models.

We show a conceptually plausible real-world scenario in Fig. 1. Given the observed trajectories of humans in the scene, a social predictor forecasts the future positions reasonably without collision. However, by adding a small perturbation of less than 5 cm to the observation trajectory, unexpectedly, a collision between predictions of agents occurs which indicates a non-complete social understanding by the predictors. The trajectories in that figure comes from S-LSTM (Alahi et al., 2016).

In this work, in contrast to the common adversarial attacks which are designed for classifiers (Ortiz-Jimenez et al., 2020; Szegedy et al., 2013), we design attack for the trajectory prediction problem which is a multimodal regression task. We use adversarial examples to study the collision avoidance behavior of the trajectory prediction models. More specifically, we investigate the worst-case social behavior of a prediction model under small perturbations of the inputs. This study has two primary motivations; (1) it is an evaluation method for the previously-proposed predictors. Our method brings counter-examples in which the models fail in having social behavior, *i.e.*, it cannot avoid collision. (2) leveraging adversarial examples, one can train models with better collision-avoidance. Furthermore, our study highlights practical concerns for employing such models in real-world applications. Notably, it is shown that state-of-the-art localization algorithms give on-average more than 0.2 m errors on human location detection at each frame (Deng et al., 2020; Bertoni et al., 2021). While our work focuses on model failures under adversarial settings, it motivates further studies of the model's performance when localization algorithms' error distributions are concerned.

We propose an adversarial attack to fool/fail the trajectory prediction models by causing collision between two agents' *predicted trajectories*. Namely, the adversarial attack aims at finding small perturbations that lead to a collision. The collision can hypothetically happen between any two agents and at any prediction timestep. However, from the attack algorithm's perspective, the choice of the agent and the timestep impacts the final perturbation's size significantly. To address that, we introduce an attention-guided adversarial attack, named *Socially-ATTended ATTAck (S-ATTack)*, which learns the best collision points. We present two variants of our attack: hard-attention and soft-attention. Our experiments demonstrate that our novel attack can find perturbations that make state-of-the-art trajectory prediction models generate wrong predictions, leading to collisions with small perturbations. We show that the achieved perturbations for one predictor can be transferred to other predictors revealing common weaknesses among models. Also, we demonstrate that the models are over-dependent on the last observation points which makes the models vulnerable. Lastly, we introduce an adversarial training scheme to make trajectory prediction models more robust. In particular, we show how our method can improve the models' social understanding in terms of collision avoidance. To the best of our knowledge, this is the first work addressing the adversarial vulnerability and robustness of trajectory prediction models. Our main contributions are summarized as follows:

- We introduce S-ATTack to assess the social understanding of the state-of-the-art trajectory prediction models.
- Our experiments shed light on the weaknesses of prediction models from different aspects.
- We demonstrate how to improve the robustness properties of the predictors using our S-ATTack.

2. Related work

Human trajectory prediction. Social-force model (Helbing and Molnar, 1998) was one of the key hand-crafted methods proposed to capture human social behaviors. They model interactions between pedestrians by means of social fields determined by repulsive and attractive forces. Social interactions have been addressed from other perspectives such as discrete choice modeling (Antonini et al., 2006), and cellular automaton model (Burstedde et al., 2001; Vizzari et al., 2015). While all these hand-crafted methods have nice interpretability and are not data-heavy, it is shown that they are not able to effectively model long-term dependencies or complex interactions (Alahi et al., 2017).

Social-LSTM (Alahi et al., 2016) was the first work that proposed the use of data-driven neural networks instead of hand-crafted functions to learn interaction dynamics between agents in the scene. Many works pursued the use of deep learning and proposed diverse approaches for learning interactions. Different types of pooling social features are studied in Kothari et al. (2021), Bartoli et al. (2018), Ivanovic and Pavone (2019), Alahi et al. (2016) to share features of agents leading to a social-compliant prediction. Convolutional neural networks were studied to process pooled features of agents in the scene (Nikhil and Tran Morris, 2018). In order to better detect relative importance of each agent, attention networks were employed in Vemula et al. (2018). Recently many works leveraged graph neural networks to model relations between agents (Kosaraju et al., 2019; Li et al., 2020; Yu et al., 2020; Mohamed et al., 2020). The multimodal distribution of data is learned by using generative adversarial networks (Gupta et al., 2018; Sadeghian et al., 2019; Amirian et al., 2019). While all mentioned works improved the performance on average and final displacement errors, they occasionally output unacceptable solutions (e.g., collisions). In this work, we attack the state-of-the-art deep learning-based recent models and reveal their weaknesses against small perturbations which challenges their social behavior.

Previously mentioned works try to learn social behavior by observing human data, which implicitly include collision avoidance. There also exist works that explicitly teach models to avoid collisions. While most of these works address the planning problem (Zeng et al., 2019; Chen et al., 2017), some adopted the inverse reinforcement learning framework to guide the network towards collision-free trajectories (van der Heiden et al., 2019). In this work, we focus our attack on the deep learning-based trajectory predictors which are fully data-driven and do not leverage human guidance.

Adversarial attacks. Adversarial attack was first introduced in Szegedy et al. (2013) by showing how vulnerable deep image classifiers are against non-perceptible yet carefully-crafted perturbations. Formally, an adversarial perturbation is defined as the minimal perturbation R which changes the output of the given classifier f :

$$\min_R \|R\|_2 \text{ subject to } f(X + R) \neq f(X), \quad (1)$$

where X is the input image. It is shown that adding these imperceptible perturbations bounded in terms of ℓ_p norms can easily fool image classifiers (Goodfellow et al., 2014; Moosavi-Dezfooli et al., 2016; Su et al., 2019). According to the ‘‘human vision knowledge’’, adding imperceptible perturbations to an image does not change its category label. Therefore, a good model should preserve its prediction while adding these perturbations to its input.

Adversarial attacks are also generalized to assess vulnerability of models against perturbations in other domains such as natural language processing (Zhang et al., 2020; Behjati et al., 2019) and tabular data (Ballet et al., 2019). Nevertheless, to the best of our knowledge, adversarial attacks are not yet explored in the context of human trajectory prediction. Human trajectory is a temporal trajectory, hence can be seen as time-series data prediction. Different types of time-series data are studied in Fawaz et al. (2019), Karim et al. (2020), Liu et al. (2020) in classification problems. While their focus is on the classification task, we target a regression task which makes the definition of a wrong output challenging. In addition, in human trajectory sequences, the imperceptibility of the perturbations is not the main interest because here, human knowledge is to respect social behaviors (avoiding collisions) in all scenarios. Lastly, a specific challenge to our problem is the freedom in the choice of the collision point which affects the perturbation size. We address these challenges by the proposed S-ATTack approach.

3. Method

In this section, we first explain the notations and definitions. Then, we will provide the details of S-ATTack.

3.1. Formulation

3.1.1. Human trajectory prediction

Pedestrian trajectory prediction addresses a regression task with sequences as inputs and outputs. At any timestep t , the i th person/agent is represented by his/her xy-coordinates (x_t^i, y_t^i) . We denote each agents’ observation sequence for T_{obs} timesteps as X^i , a $T_{obs} \times 2$ matrix and the observation sequences of all the n agents in the scene as $X = (X^1, \dots, X^n)$. Given X , the trajectory predictor f predicts the next T_{pred} positions of all agents $Y = (Y^1, \dots, Y^n) = f(X^1, \dots, X^n)$ where Y^i is a $T_{obs} \times 2$ matrix.

3.1.2. Adversarial examples for trajectory prediction

Equipped with the notations introduced in Section 3.1.1, we will provide a definition of adversarial examples for trajectory prediction. In this paper, without loss of generality, we assume the perturbation R is added to the candidate agent which is arbitrarily chosen among agents. Note that in the experiments, all agents are considered as the candidate agent one by one. $\hat{X}^1 = X^1 + R$ while the observations of other agents (which we refer to as neighbors) $\{X^j\}_{j \neq 1}$ remain fixed. Therefore, R is a $T_{obs} \times 2$ matrix of adversarial perturbation, the adversarial example is $\hat{X} = (X^1 + R, X^2, \dots, X^n)$ and the output of the predictor for that example is $\hat{Y} = (\hat{Y}^1, \dots, \hat{Y}^n) = f(\hat{X})$. Formally, given a small constant $\epsilon > 0$, a collision distance threshold γ and the maximum of the norm of all rows of a matrix $\|\cdot\|_{\max}$, a socially-attended adversarial example is obtained if:

$$\exists t, j \neq 1, \|R\|_{\max} \leq \epsilon : \left\| \hat{Y}_t^j - \hat{Y}_t^1 \right\| < \gamma. \quad (2)$$

In other words, this type of adversarial examples is based on perturbing an observation trajectory so that f predicts the future timesteps with at least a collision (the distance less than γ) between two agents j and 1 in one timestep t . In addition, without loss of generality, we focus on the collisions between the candidate agent and neighbors. Clearly, it can directly be expanded to collisions between any two agents. In the next section, we will describe how we obtain R using Socially-attended attack.

3.2. Socially-attended attack (S-ATTack)

We propose three optimization problems based on different attention mechanisms for socially-attended attack to find suitable perturbations for a collision. The optimization problems are explained in Sections 3.2.1–3.2.3 and are solved using an iterative algorithm based on projected gradient descent elaborated in Section 3.3.

Before introducing the three optimizations, we will explain the distance matrix which is used in all three optimizations. Given the perturbation R , and a model f , we define the distance matrix $D(R) \in \mathbb{R}^{(n-1) \times T_{pred}}$ as a function of the input perturbation R . It includes the pairwise distance of all neighboring agents from the candidate agent in all prediction timesteps. Let $d_{j,t}$ denote the element at j th row and t th column of $D(R)$, i.e., the distance of the agent j from the candidate agent at timestep t of the prediction timesteps:

$$d_{j,t} := \left\| \hat{Y}_t^j - \hat{Y}_t^1 \right\|. \quad (3)$$

Hence, for a particular R , the distance matrix $D(R)$ can be leveraged to indicate whether a collision has occurred. We now explain three methods to find such a perturbation by optimizing different cost functions depending on $D(R)$.

3.2.1. No-attention loss

Our first attempt is to introduce a simple social loss to find perturbations that make collisions in the trajectory prediction sequences. To achieve that, we find the perturbations that make a collision among the predictions of humans in the prediction model. One naive solution is to minimize the sum of distances between the candidate agent and its neighbors in all prediction timesteps:

$$\min_R \|D(R)\|_F. \quad (4)$$

This naive scheme gives the same attention to all agents, which may not be efficient in obtaining small perturbations. For instance, a far agent may not be a good potential candidate for collision and should receive lower attention.

3.2.2. Hard-attention loss

A better approach to cause a collision is to target a specific agent in a specific timestep instead of taking an average over all neighboring agents. Then, the model is attacked in a way that the distance of the chosen agent with the candidate agent is decreased in the corresponding timestep until the collision occurs. The equation of the hard-attention attack is as follows:

$$\begin{aligned} & \min_{R,W} \text{Tr}(W^T D(R)) + \lambda_r \|R\|_F, \\ & \text{s.t. } w_{j,t} = \delta_{jk} \delta_{tm}, \quad k, m = \underset{j,t}{\text{argmin}} d_{j,t}, \end{aligned} \quad (5)$$

where δ is the Kronecker delta function. Besides, W is the attention weight matrix and $w_{j,t}$ is the attention weight for the agent j at timestep t of the prediction timesteps. Indeed, the associated weight to the target agent $w_{k,m}$ which is the closest agent-timestep is 1 and others are 0. The socially-attended loss is the trace (Tr) of multiplication of $D(R)$ by the transpose of W added with the regularization on the perturbation with the balancing coefficient λ_r that encourages finding a small perturbation sequence to make a collision.

3.2.3. Soft-attention loss

The main drawback of Eq. (5) is that the target point (k, m) for a collision is selected based on the assumption that attacking the closest agent-timestep requires small-enough perturbation. This confines the target point selection and might not find the most optimum target point. Note that the models are non-linear and a collision with the closest agent-timestep may not essentially require the smallest perturbation.

To address that, we let the attack attend to the optimal target by itself. We introduce a soft-attention mechanism in which the weights associated to each agent-timestep is assigned by the attack in order to achieve a smaller perturbation. The equation of the soft-attention attack is as follows:

$$\begin{aligned} & \min_{R,W} \text{Tr}(W^T \tanh(D(R))) + \lambda_r \|R\|_F - \lambda_w \|W\|_F, \\ & \text{s.t. } \sum_{j,t} w_{j,t} = 1, \quad w_{j,t} \geq 0, \end{aligned} \quad (6)$$

where \tanh is applied to the entries of $D(R)$ in order to concentrate less on very far agent-timesteps. Also, we discourage uniformity of weights by subtracting the Frobenius norm of W multiplied by a scalar λ_w .

W is initialized with a uniform distribution. It is progressively updated and puts more weights to the more probable targets for making a collision. Near the convergence point, the best target agent receives a weight value close to 1 while the rest receive 0. Note that W and R are optimized jointly per input sample. We show one example of how W changes in the training in the supplementary material.

We will compare our social adversarial attacks (hard-attention Eq. (5) and soft-attention Eq. (6)) in Section 4.3. For brevity, in the rest of the paper, by S-ATTack, we refer to the attack with soft-attention.

3.3. S-ATTack algorithm

In this section, we explain how adversarial perturbations are achieved using the introduced loss functions. The pseudo-code of the algorithm is written in Algorithm 1. The method is an iterative algorithm with maximum k_{max} iterations. At each iteration, first, perturbed inputs \hat{X} are calculated by adding perturbation R to X^1 . Then, we find new predictions \hat{Y} using the perturbed inputs \hat{X} . Next, we use projected gradient descent (Madry et al., 2017) to solve the constraint optimization problems introduced in the previous subsections. Namely, the perturbation R is updated using the gradient of the equations defined above with hyperparameter α and then projected to the ℓ_∞ ball with radius ϵ . Finally, if a collision in the predictions exists, the algorithm stops otherwise it continues until the maximum iterations.

Algorithm 1: The pseudo-code of S-ATTack algorithm

Input: Sequence X , Predictor f
Output: Perturbed sequence \hat{X}

```

1 Initialize  $k \leftarrow 0, \hat{X} \leftarrow X, R \leftarrow 0$ 
2 while  $k < k_{max}$  do
3    $\hat{X} = (X^1 + R, X^2, \dots, X^n)$ 
4    $\hat{Y} = f(\hat{X})$ 
5    $R = R + \alpha \nabla (Eq. (4) \text{ or } Eq. (5) \text{ or } Eq. (6))$ 
6    $[R_{i,j}] = [\min(R_{i,j}, \epsilon)]$ 
7   compute  $D$  using Eq. (3)
8   if  $\exists s \neq t : d_{s,t} < \gamma$  then
9     | return  $\hat{X}$ 
10  end
11   $k = k + 1;$ 
12 end

```

4. Experiments

We conduct the experiments to answer the following questions: (1) How vulnerable are the trajectory prediction models against perturbations on the input sequence? (2) Which of the proposed socially-attended attacks can cause collisions more successfully with smaller perturbations? (3) Are we able to leverage the adversarial examples to improve the robustness of the model? Do they help in better learning the social behavior?

4.1. Experimental setup

4.1.1. Baselines

In order to show the effectiveness of our attack, we conduct our experiments on six well-established trajectory prediction models.

- **Social-LSTM** (Alahi et al., 2016) (**S-LSTM**): where a social pooling method is employed to model interactions based on shared hidden states of LSTM trajectory encoders.
- **Social-Attention** (Vemula et al., 2018) (**S-Att**): where a self-attention block is in charge of learning interactions between agents.
- **Social-GAN** (Gupta et al., 2018) (**S-GAN**): where a max-pooling function is employed to encode neighborhood information. They leverage a generative adversarial network (GAN) to learn the distribution of trajectories.
- **Directional-Pooling** (Kothari et al., 2021) (**D-Pool**): where the features of each trajectory is learned using the relative positions as well as the relative velocity and then pool the learned features to learn social interactions.
- **Social-STGCNN** (Mohamed et al., 2020) (**S-STGCNN**): where graph convolutional neural network is employed to learn the interactions.
- **PECNet** (Mangalam et al., 2020) (**PECNet**): where a self-attention based social pooling layer is leveraged with a variational auto-encoder (VAE) network.

4.1.2. Datasets

ETH (Pellegrini et al., 2010), **UCY** (Lerner et al., 2007), and **WildTrack** (Chavdarova et al., 2018): These are well-established datasets with human positions in world-coordinates. We employ two variants of these datasets for our experiments: (1) for S-LSTM, S-Att, S-GAN and D-Pool baselines, we utilized Trajnet++ (Kothari et al., 2021) benchmark which provides identical data splits and data pre-processing. The observation and prediction lengths are considered as 9 and 12, respectively. (2) Since S-STGCNN official code contains its specific pre-processing and data-split on ETH and UCY, we employed the released code to be consistent with their official implementation. Here, the observation and prediction lengths are considered as 8 and 12, respectively.

SDD (Robicquet et al., 2016): The Stanford Drone Dataset is a human trajectory prediction dataset in bird's eye view. PECNet is one of the state-of-the-art methods with official published code on this dataset. Hence, we report PECNet performance on this dataset. The observation and prediction lengths are considered as 8 and 12, respectively.

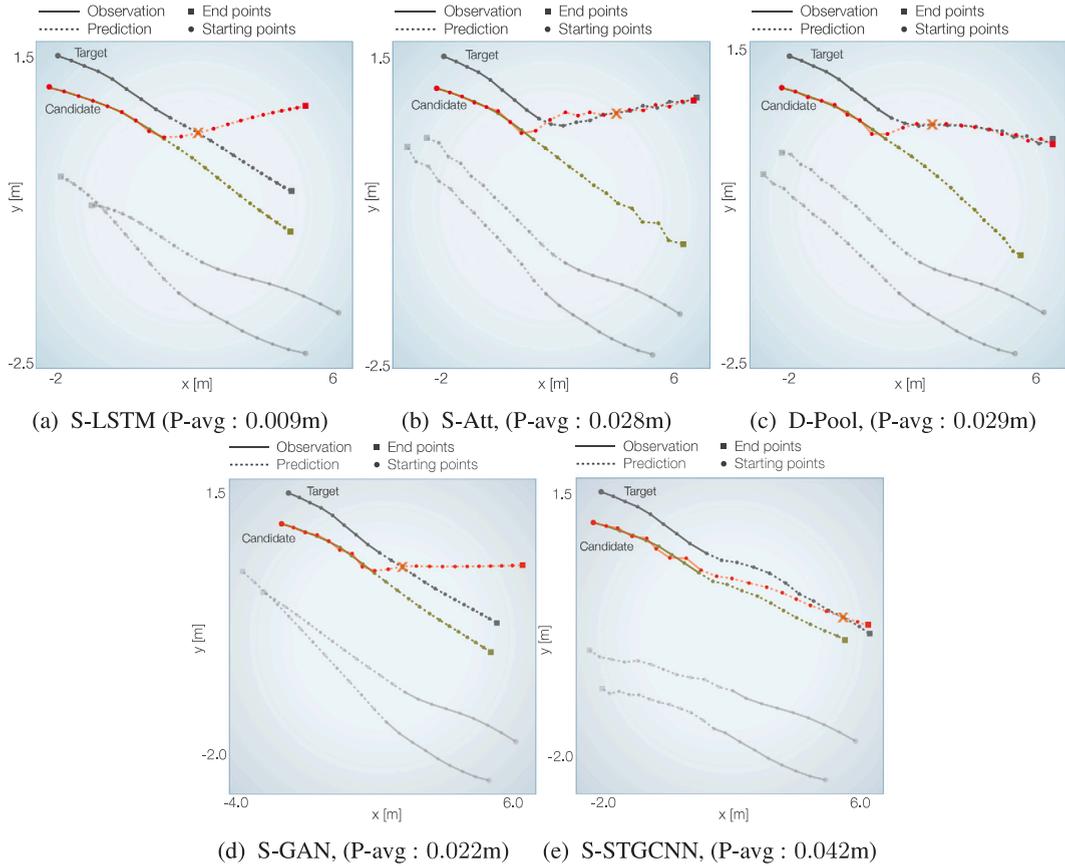


Fig. 2. Comparison of the performance of different models under our attack. The candidate agent is depicted with green before the perturbation and red after it. For brevity, we have not shown the prediction of neighboring agents before the attack. The scale of y axis is enlarged to better show the difference. The orange X denotes the target point. Our attack achieves collisions by adding small perturbations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.1.3. Implementation details

We set the maximum number of iterations to 100. Inspired by the localization algorithm errors mentioned in the introduction, the maximum size of the perturbation for each point ϵ is considered 0.2 m. Also, we set λ_r and λ_w in Eq. (5) and Eq. (6) equal to 0.1 and 0.5 respectively. The full list of hyperparameters will be provided in the supplementary material.

4.1.4. Metrics

In the experiments, we report the performances according to the following metrics:

- **Collision Rate (CR):** this metric measures the existence of collision in the predicted trajectories of the model. Indeed, it calculates the percentage of samples in which at least one collision in the predicted trajectories between the candidate agent and its neighbors occurs. This metric assesses whether the model learned the notion of collision avoidance. Note that we set the distance threshold for indicating a collision γ in Eq. (2) equal to 0.2 m.
- **Perturbation average (P-avg):** the average of perturbation sizes at each timestep which is added to the input observation. The numbers are reported in meters.
- **Average/Final Displacement Error (ADE/FDE):** the average/final displacement error between the predictions of the model and the ground-truth values. This metric is commonly used to report the performance of trajectory prediction models and is reported in meters.

4.2. Attack results

We first provide the quantitative results of applying S-ATTack to the baselines in Table 1. The results indicate a substantial increase in the collision rate (at least 3 times) across all baselines by adding perturbations with P-avg smaller than 0.11 m. This questions the social behavior of the models in terms of collision avoidance.

Table 1

Comparing the performance of different baselines before (Original) and after the attack (Attacked). Horizontal lines separate models with different datasets.

Model	Original	Attacked	
	CR [%] ↓	CR [%] ↓	P-avg [m] ↓
S-LSTM (Alahi et al., 2016)	7.8	89.8	0.031
S-Att (Vemula et al., 2018)	9.4	86.4	0.057
S-GAN (Gupta et al., 2018)	13.9	85.0	0.034
D-Pool (Kothari et al., 2021)	7.3	88.0	0.042
S-STGCNN (Mohamed et al., 2020)	16.3	59.1	0.11
PECNet (Mangalam et al., 2020)	15.0	64.9	0.071

Table 2

Comparing different proposed attack methods on D-Pool.

Attacks	CR [%] ↓	P-avg [m] ↓
Random noise	17.6	0.199
No-attention Eq. (4)	44.7	0.179
Hard-attention Eq. (5)	84.8	0.041
Soft-attention Eq. (6)	88.0	0.042

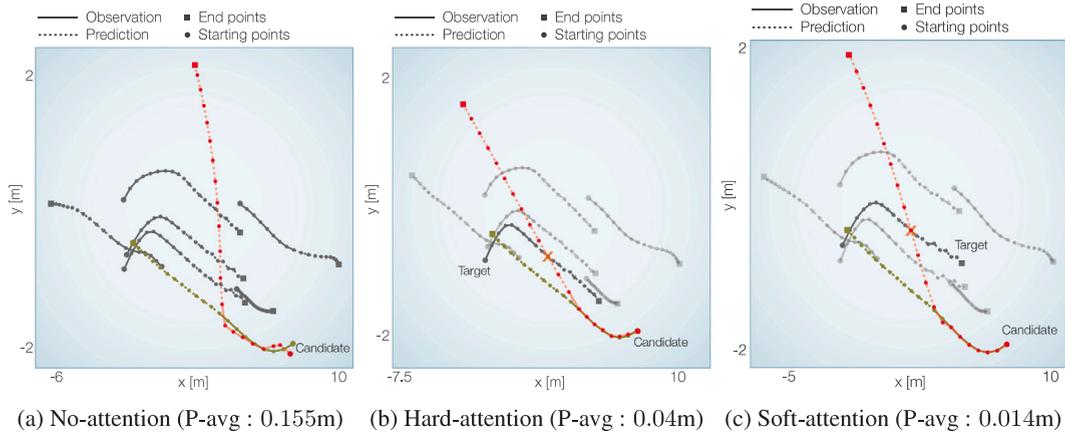


Fig. 3. Comparison of different attack approaches. The no-attention approach Eq. (4) could not make a collision despite large P-avg. The hard-attention approach Eq. (5) led to a collision with a smaller P-avg. A better target point for collision is found by soft-attention Eq. (6) leading to a collision with the least P-avg.

Fig. 2 visualizes the performance of the baselines S-LSTM, S-Att, D-Pool, S-GAN and S-STGCNN under our attack with the same input. Note that all the baselines are trained on the first group of datasets in 4.1.2. S-LSTM does not change its predictions after adding perturbations but S-Att and D-Pool counteract to avoid collisions. This shows that there exists some collision avoidance behavior understanding in some prediction models, but they are not enough for avoiding a collision. We show more successful cases and also some failure cases in the supplementary material.

As D-pool performs better than others in terms of collision avoidance before attack, in the rest of the paper, we conduct our main experiments on it.

4.3. Comparison of different attention methods

In this section, we compare how different strategies for choosing the collision point affects the collision rate and the perturbation size. The quantitative results are shown in Table 2. We report the results of Gaussian random noise with variance of 0.2, no-attention Eq. (4), hard-attention Eq. (5), and soft-attention Eq. (6). Random noise and no-attention approaches have small collision rates with large perturbation sizes which indicates the need for a strategy for selecting the collision point. Both hard-attention and soft-attention mechanisms substantially improve collision rate. However, the freedom in the soft-attention approach lets it smartly find better collision points in some of the samples leading to a higher collision rate. Fig. 3 visualizes the performance of different attack approaches for one data sample. In the illustrated example, in contrast to other two approaches, Soft-attention targets a collision point which is further but easier to collide, thus, leads to a smaller perturbation size.

Table 3

Studying the transferability of adversarial examples. The adversarial examples are learned for source models and are transferred to the target models for the evaluation. The reported numbers are Collision Rate (CR) values.

Target models	Source models			
	S-LSTM	S-Att	S-GAN	D-Pool
S-LSTM (Alahi et al., 2016)	89.8	82.7	85.1	68.3
S-Att (Vemula et al., 2018)	53.0	86.4	57.8	70.0
S-GAN (Gupta et al., 2018)	40.8	59.7	85.0	84.1
D-Pool (Kothari et al., 2021)	88.4	81.3	55.6	88.0
S-Forces (Helbing and Molnar, 1998)	0.69	0.70	1.30	0.60

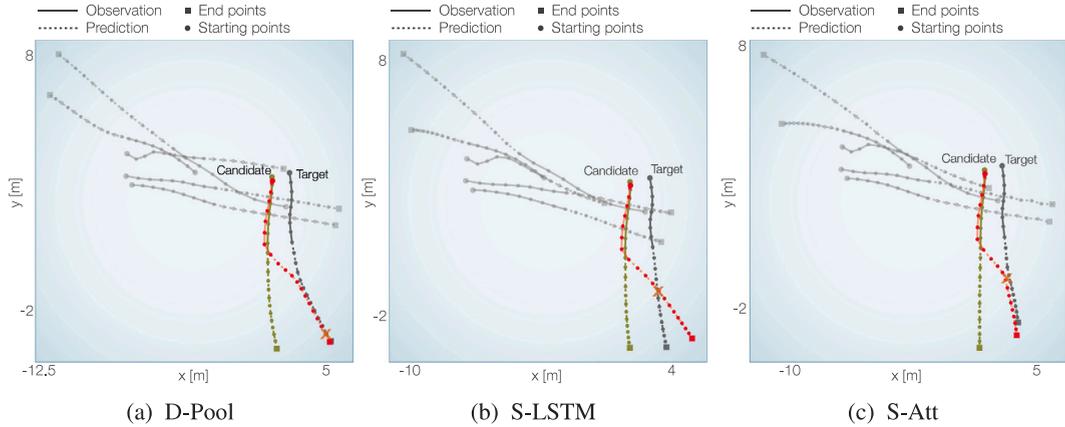


Fig. 4. Transferring an adversarial example obtained from attacking D-Pool model to S-LSTM and S-Att models. P-avg is 0.151 m. We can observe that the perturbation generates collisions in both S-LSTM and S-Att models, although not optimized for them.

4.4. Transferability

A common practice in adversarial attack studies is to investigate the transferability of perturbations achieved for one model across other models. This shows the existence of common weaknesses across different models or biases in the dataset leading to brittle features in the models. Note that due to the inconsistency of the data used for different baselines, we perform this study only for the models which use the Trajnet++ data.

To study how the generated perturbations can transfer to other models, source models were attacked and the achieved perturbations were used to evaluate others as target models. In addition to the data-driven models, we study the performance of a physics-based model, Social-forces (S-Forces) (Helbing and Molnar, 1998) against generated perturbations. Note that S-Forces has zero collision rate before the attack.

Table 3 shows that the transferred perturbations can make substantial collision rates for the data-driven models. This means that there exist common defects across different models. Expectedly, S-Forces is robust against the perturbations. This is due to the fact that collision avoidance is an explicit rule defined for the model. However, this robustness comes with the cost of accuracy loss as reported in Alahi et al. (2016). We will further analyze our findings in Section 4.6. Fig. 4 shows the result of one identical perturbation on three different models.

4.5. Enhancing the social understanding

We utilize our S-ATTack to improve the collision avoidance of the model. To this end, we employ a similar approach to Madry et al. (2017). We fine-tune the model using a combination of the original training data and the adversarial examples generated by our S-ATTack method. In this experiment, we set the maximum perturbation size ϵ equal to 0.03.

Table 4 indicates that the model's collision avoidance could improve by 11%. Moreover, the collision rate after attack improves by 60% meaning that it is much less vulnerable to the attack. As shown in the table, fine-tuning the model with random noise could not improve the collision avoidance. Therefore, we conclude that our adversarial examples provide useful information to improve the collision avoidance of the model. Note that the prediction error of the model in terms of ADE/FDE is slightly increased. This shows a trade-off between accuracy and robustness. This can be similar to the findings in the previous works on image classifiers (Tsipras et al., 2018).

We also show the performance of these models visually. In Fig. 5(a), the original model's prediction with an occurring collision is seen while the enhanced model prediction is collision-free (Fig. 5(b)). In the next figure, Fig. 6(a), we can observe that attacking D-Pool leads to a collision with a P-avg of 6.6 but the enhanced model is robust against attack (Fig. 6(b)) indicating a better collision avoidance understanding.

Table 4

Comparing the original model and the fine-tuned model with random-noise data augmentation (D-Pool w/ rand noise) and S-ATTack adversarial examples (D-Pool w/ S-ATTack). The numbers are on Trajnet++ challenge testset. ADE, FDE are reported in meters.

	ADE/FDE [m] ↓	Original		Attacked	
		CR [%] ↓	CR gain [%] ↑	CR [%] ↓	CR gain [%] ↑
D-Pool	0.57/1.23	7.3	–	37.3	–
D-Pool w/ rand noise	0.57/1.23	7.5	–2.7	36.1	+3.2
D-Pool w/ S-ATTack	0.60/1.28	6.5	+10.4	14.7	+60

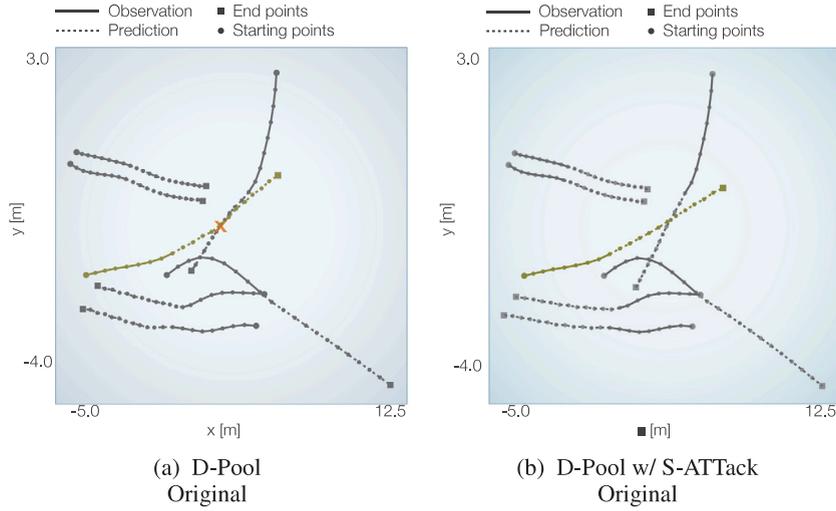


Fig. 5. Comparison of the performance of the original model and the enhanced one with S-ATTack. The original model has a collision in its predictions (a) while the enhanced model prediction is collision-free (b).

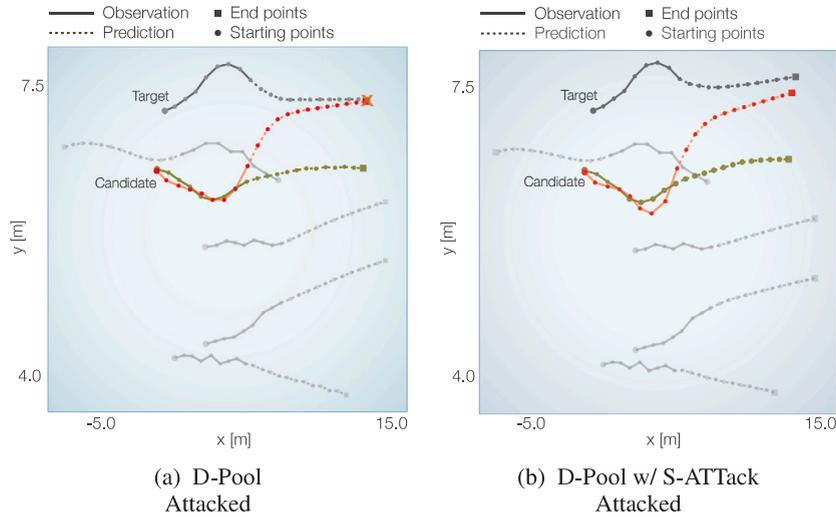


Fig. 6. Comparison of the vulnerability of the victim model under attack and the enhanced one with S-ATTack. While D-Pool can be attacked to cause a collision with a P-avg of 6.6 (a), the enhanced model cannot be attacked even with a large perturbation (b).

4.6. Discussions

In this section, equipped with the performed experiments in previous parts, we want to shed light on the weaknesses of the studied trajectory prediction models. Also, we will mention the limitations of our work.

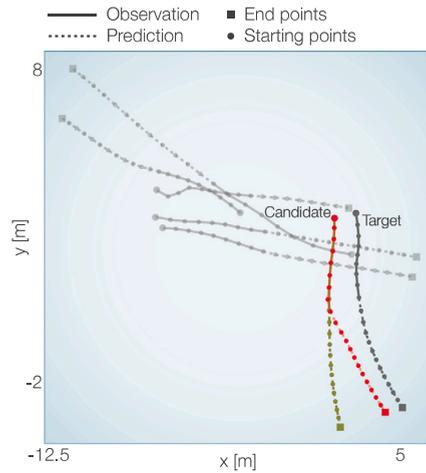


Fig. 7. We froze the predictions of neighboring agents for the attack algorithm. The algorithm fails in making collision since it is unaware of the counteracts of neighbors to the perturbations. This indicates that the prediction model counteracts with the perturbations. Also, the attack algorithm requires to consider other agent's counteracts to cause a collision. The same sample was successfully attacked in Fig. 4.

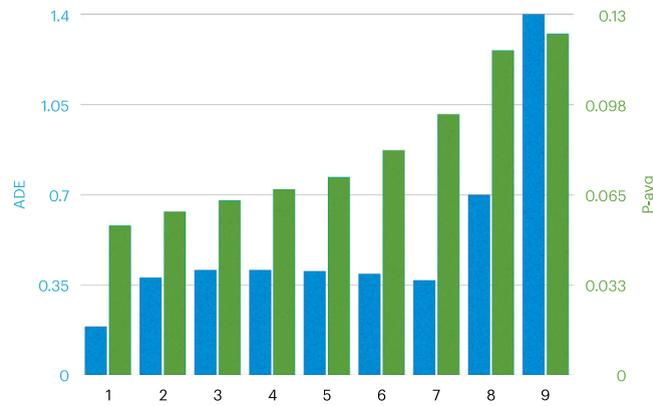


Fig. 8. Two overlaid figures analyzing the impact of different timesteps. The blue is the average change in the predictions by adding a random noise of size 0.2 m to different observation timesteps. The green curve shows the perturbation size at each timestep in perturbations found by S-ATTack. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

1. Although the trajectory prediction models are designed to capture interactions among people, our results in Section 4.2 showed that they are fragile and cannot generalize to perturbed data. Moreover, we showed in Section 4.4 that the perturbations are transferable across different models indicating the existence of common weaknesses across these models. We have two hypotheses as the reasons of high vulnerability to the attack and transferability of the perturbations: (1) this can be due to the lack of collision avoidance inductive bias in the models. While inductive bias is more influential when the size of the training data is limited, it may be of less impact in large data regime. (2) because of limited training data, there exists unexplored input space for the predictor model. We leave a detailed study on these points for future work.
2. The success of our attack in making collisions may raise the concern that the models are actually unaware of the social interactions. We perform an experiment to assess this point. We attack the model while keeping the predictions of neighboring agents frozen for the attack algorithm. Therefore, the attack cannot consider other agents' counteracts to the perturbations. As shown in Fig. 7, the attack fails in causing collision. This result indicates that while S-ATTack can cause collisions in the models' predictions, the models have limited social understanding allowing them to counteract perturbations.
3. To compare the models' sensitivity to the perturbations added to different observation points, we add small random noise (less than 0.2 m) to each point separately and measure its effect on the prediction error. Fig. 8 shows that the predictors are over-dependent on the last observation point. It is consistent with our findings that the perturbations generated by S-ATTack tend to have larger components on the last timesteps trying to use this feature.
4. Are different prediction timesteps equally vulnerable to the collisions? To answer this question, we consider a scenario where two pedestrians are crossing each other at different prediction timesteps and measure the collision rate. We use S-Forces, S-LSTM and D-Pool. Fig. 9(a) visualizes the predictions of S-LSTM and S-Forces models for one scenario. While S-LSTM is unable to avoid the collision, S-Forces decreases the prediction speed in the first points to avoid the collision. Then, we study the

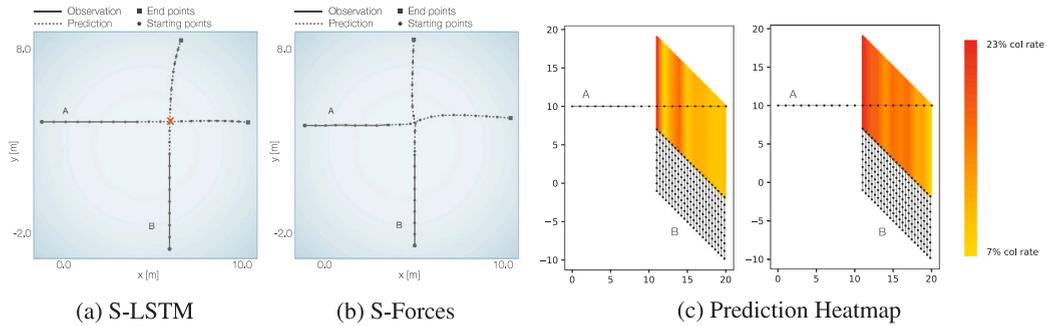


Fig. 9. Collision rate analysis for different timesteps. We cross two pedestrians A,B to see if the prediction models can avoid the facing collision. (a,b) We visualize the performance of S-LSTM and S-Forces models in a scenario where two agents cross in the third prediction timestep. S-LSTM is not avoiding the collision whereas S-Forces avoids it due to the hand-crafted knowledge of collision avoidance. (c) Analysis of collision rates of two data-driven model, D-Pool (left) and S-LSTM (right) on different timesteps. We cross two pedestrians A,B on different timesteps and report the collision rate for each of them. Dark yellow is the observation sequence for both agents and green is the prediction of A. The red and yellow colors show high and low collision rates, respectively. We observe different patterns for each model. We do not visualize the first two prediction timesteps as the model is unable to change the manoeuvre instantly. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5

Leveraging smoothing function in the attack. Smoothing in defense makes the perturbed observation smooth before feeding to the predictor which can be considered as a defense method against the attack. S-attack with smoothing includes the smoothing function in the optimization problem to achieve a smoothed perturbation.

Attacks	CR [%] ↓
D-pool	7.3
D-pool + S-attack	88.0
D-pool + S-attack + Smoothing in defense	61.6
D-pool + S-attack with smoothing + Smoothing in defense	72.0

sensitivity of models in different timesteps with regards to collision avoidance. To this end, we run scenarios like Fig. 9(a),(b) 1000 times for each timestep where small Gaussian noise is added to the observations. The result are shown in a heatmap in Fig. 9(c). First, a high collision rate is observed for both models revealing their weaknesses in capturing agent-agent interactions. Second, each model has a different pattern of sensitivity with respect to the timestep. Initial timesteps have higher collision rate since it is more difficult to change the manoeuvre in the first timesteps. While smaller collision rates are expected for later timesteps, in both models, the collision rate does not monotonically decrease. This shows that the models are more vulnerable in the middle timesteps. This can be due to the biases in the data or model structure.

- Our attack perturbs only the candidate agent to achieve a collision. The same method can be employed to perturb other agents. Fig. 10 shows three different scenarios in a scene where a collision occurs by perturbing different agents. However, the collision type of Fig. 10(c) is not considered in our method and can be addressed as a future work.
- The perturbations are calculated through a joint optimization problem and there is no constraint on the smoothness of the perturbations. To study the impact of our adversarial attack in the presence of a smoothing filter, a 4th order polynomial is fit to smooth the trajectories according to Becker et al. (2018). In the first and second rows of Table 5, the performance of the original model and with S-attack is observed. Next, we add the smoothing in defense (3rd row) and in both defense and attack (4th row). Smoothing in defense makes the final calculated perturbations smooth before feeding to the predictor. Smoothing the trajectory reduces collision but still the attack is highly effective. In the third experiment, we include the smoothing in the attack optimization algorithm where the perturbation is smoothed after each optimization iteration. Therefore, S-attack finds a smooth perturbation which shows higher collision rate compared to the previous experiment. Note that CR is lower than the original experiment which is because of the additional polynomial constraint enforced that reduces the search space.

5. Conclusion and future works

In this work, we studied the robustness properties of trajectory prediction models in terms of social understanding under adversarial attack. We introduce our Socially-ATTended ATTack (S-ATTack) to cause collisions in state-of-the-art prediction models with small perturbations. Adversarial training using S-ATTack can not only make the models more robust against adversarial attacks, but also reduce the collision rate and hence, improve their social understanding. This paper sheds light on the common weaknesses of trajectory prediction models opening a window towards their social understandings.

Our work is a first step that highlights the lack of social understanding in the models. The attack method can be extended by considering the velocity that a collision occurs with, or considering other notions of social behavior such as grouping. To approach socially-aware predictors, we believe that the field lacks two main components. First, using more socially-related metrics instead

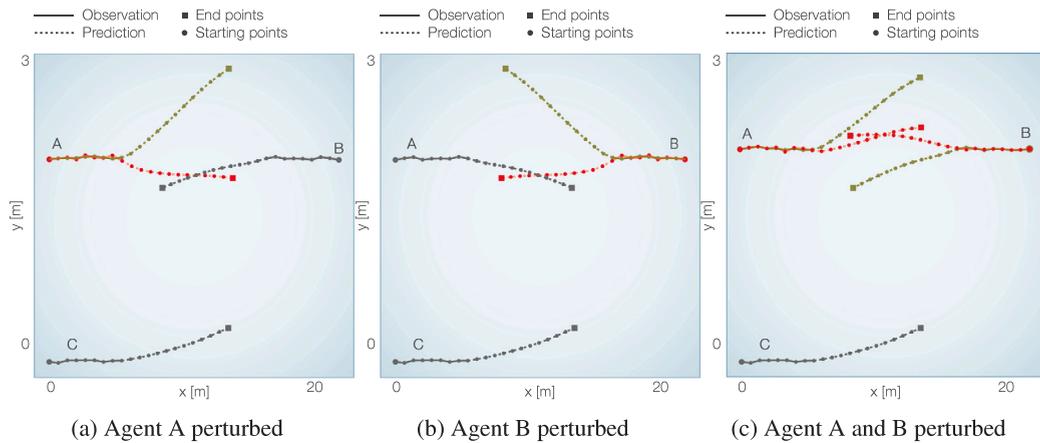


Fig. 10. Perturbing different agents in a scenario to achieve collision between agents A and B. (a) Perturbing A which is the candidate agent. (b) Perturbing the agent B. (c) Perturbing both A and the other agent B that is colliding with. In all cases, a collision between A and B is predicted. This shows that S-attack is effective for other problem formulations.

of ADE/FDE for training and evaluating. This is an area of research where metrics like the time-to-collision metric (Karamouzas et al., 2014), are required to be studied more. Second, supervised learning in the absence of proper inductive biases cannot learn social behaviors properly especially with the current small-scale datasets. One direction to address the challenge is to combine physics-based models and neural networks. Also, having more challenging and large-scale datasets can help the models learn the bias by themselves.

CRediT authorship contribution statement

Saeed Saadatnejad: Conceptualization, Methodology, Software, Investigation, Validation, Visualization, Writing. **Mohammadhossein Bahari:** Conceptualization, Methodology, Software, Investigation, Validation, Visualization, Writing. **Pedram Khorsandi:** Software, Investigation, Validation, Visualization. **Mohammad Saneian:** Software, Investigation, Validation, Visualization. **Seyed-Mohsen Moosavi-Dezfooli:** Methodology, Supervision, Writing. **Alexandre Alahi:** Methodology, Supervision, Writing, Resources, Project administration.

Acknowledgments

This project was funded by Honda R&D Co., Ltd and from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 754354.

References

- Alahi, Alexandre, Goel, Kratharth, Ramanathan, Vignesh, Robicquet, Alexandre, Fei-Fei, Li, Savarese, Silvio, 2016. Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 961–971.
- Alahi, Alexandre, Ramanathan, Vignesh, Goel, Kratharth, Robicquet, Alexandre, Sadeghian, Amir A, Fei-Fei, Li, Savarese, Silvio, 2017. Learning to predict human behavior in crowded scenes. In: Group and Crowd Behavior for Computer Vision. Elsevier, pp. 183–207.
- Amirian, Javad, Hayet, Jean-Bernard, Pettré, Julien, 2019. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.
- Antonini, Gianluca, Bierlaire, Michel, Weber, Mats, 2006. Discrete choice models of pedestrian walking behavior. *Transp. Res. B* 40 (8), 667–687.
- Bahari, Mohammadhossein, Nejjar, Ismail, Alahi, Alexandre, 2021. Injecting knowledge in data-driven vehicle trajectory predictors. *Transp. Res. C* 128, 103010.
- Ballet, Vincent, Renard, Xavier, Aigrain, Jonathan, Laugel, Thibault, Frossard, Pascal, Detyniecki, Marcin, 2019. Imperceptible adversarial attacks on tabular data. arXiv preprint arXiv:1911.03274.
- Bartoli, Federico, Lisanti, Giuseppe, Ballan, Lamberto, Del Bimbo, Alberto, 2018. Context-aware trajectory prediction. In: 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, pp. 1941–1946.
- Becker, Stefan, Hug, Ronny, Hübner, Wolfgang, Arens, Michael, 2018. An evaluation of trajectory prediction approaches and notes on the trajnet benchmark. arXiv preprint arXiv:1805.07663.
- Behjati, Melika, Moosavi-Dezfooli, Seyed-Mohsen, Baghshah, Mahdieh Soleymani, Frossard, Pascal, 2019. Universal adversarial attacks on text classifiers. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 7345–7349.
- Bennewitz, Maren, Burgard, Wolfram, Thrun, Sebastian, 2002. Learning motion patterns of persons for mobile service robots. In: Proceedings of the IEEE International Conference on Robotics and Automation. Vol. 4, IEEE, pp. 3601–3606.
- Bertoni, Lorenzo, Kreiss, Sven, Mordan, Taylor, Alahi, Alexandre, 2021. Monstereo: When monocular and stereo meet at the tail of 3D human localization. In: International Conference on Robotics and Automation (ICRA).
- Blue, Victor J., Adler, Jeffrey L., 2001. Cellular automata microsimulation for modeling bi-directional pedestrian walkways. *Transp. Res. B* 35 (3), 293–312.
- Bouhsain, Smail Ait, Saadatnejad, Saeed, Alahi, Alexandre, 2020. Pedestrian intention prediction: A multi-task perspective. *Eur. Assoc. Res. Transp. (HEART)*.

- Burstedde, Carsten, Klauck, Kai, Schadschneider, Andreas, Zittartz, Johannes, 2001. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A* 295 (3–4), 507–525.
- Chavdarova, Tatjana, Baqué, Pierre, Bouquet, Stéphane, Maksai, Andrii, Jose, Cijo, Bagautdinov, Timur M., Lettry, Louis, Fua, Pascal, Gool, Luc Van, Fleuret, François, 2018. Wildtrack: A multi-camera HD dataset for dense unscripted pedestrian detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 5030–5039.
- Chen, Yu Fan, Liu, Miao, Everett, Michael, How, Jonathan P., 2017. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 285–292.
- Deng, Wenlong, Bertoni, Lorenzo, Kreiss, Sven, Alahi, Alexandre, 2020. Joint human pose estimation and stereo 3D localization. In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2324–2330.
- Duives, Dorine C., Daamen, Winnie, Hoogendoorn, Serge P., 2013. State-of-the-art crowd motion simulation models. *Transp. Res. C* 37, 193–209.
- Fawaz, Hassan Ismail, Forestier, Germain, Weber, Jonathan, Idoumghar, Lhassane, Muller, Pierre-Alain, 2019. Adversarial attacks on deep neural networks for time series classification. In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1–8.
- Goodfellow, Ian J., Shlens, Jonathon, Szegedy, Christian, 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- Gupta, Agrim, Johnson, Justin, Fei-Fei, Li, Savarese, Silvio, Alahi, Alexandre, 2018. Social gan: Socially acceptable trajectories with generative adversarial networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 2255–2264.
- Helbing, Dirk, Molnar, Peter, 1998. Social force model for pedestrian dynamics. *Phys. Rev. E* 51.
- Ivanovic, Boris, Pavone, Marco, 2019. The trajectory: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. pp. 2375–2384.
- Karamouzas, Ioannis, Skinner, Brian, Guy, Stephen J., 2014. Universal power law governing pedestrian interactions. *Phys. Rev. Lett.* 113 (23), 238701.
- Karim, Fazle, Majumdar, Somshubra, Darabi, Houshang, 2020. Adversarial attacks on time series. *IEEE Trans. Pattern Anal. Mach. Intell.*
- Kosaraju, Vineet, Sadeghian, Amir, Martín-Martín, Roberto, Reid, Ian, Rezaatoghhi, Hamid, Savarese, Silvio, 2019. Social-bi-gat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In: *Advances in Neural Information Processing Systems (NeurIPS)*. pp. 137–146.
- Kothari, Parth, Kreiss, Sven, Alahi, Alexandre, 2021. Human trajectory forecasting in crowds: A deep learning perspective. *IEEE Trans. Intell. Transp. Syst.*
- Lerner, Alon, Chrysanthou, Yiorgos, Lischinski, Dani, 2007. Crowds by example. *Comput. Graph. Forum* 26, 655–664.
- Li, Jiachen, Ma, Hengbo, Zhang, Zhihao, Tomizuka, Masayoshi, 2020. Social-wagdat: Interaction-aware trajectory prediction via wasserstein graph double-attention network. *arXiv preprint arXiv:2002.06241*.
- Liu, Jian, Akhtar, Naveed, Mian, Ajmal, 2020. Adversarial attack on skeleton-based human action recognition. *IEEE Trans. Neural Netw. Learn. Syst.*
- Madry, Aleksander, Makelov, Aleksandar, Schmidt, Ludwig, Tsipras, Dimitris, Vladu, Adrian, 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.
- Mangalam, Karttikeya, Girase, Harshayu, Agarwal, Shreyas, Lee, Kuan-Hui, Adeli, Ehsan, Malik, Jitendra, Gaidon, Adrien, 2020. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 759–776.
- Mohamed, Abdullallah, Qian, Kun, Elhoseiny, Mohamed, Claudel, Christian, 2020. Social-stgcn: 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 (CVPR)*. pp. 14424–14432.
- Moosavi-Dezfooli, Seyed-Mohsen, Fawzi, Alhussein, Frossard, Pascal, 2016. Deepfool: a simple and accurate method to fool deep neural networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 2574–2582.
- Nikhil, Nishant, Tran Morris, Brendan, 2018. Convolutional neural network for trajectory prediction. In: *European Conference on Computer Vision (ECCV) Workshops*.
- Ortiz-Jimenez, Guillermo, Modas, Apostolos, Moosavi, Seyed-Mohsen, Frossard, Pascal, 2020. Hold me tight! influence of discriminative features on deep network boundaries. In: *Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (Eds.), Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33, Curran Associates, Inc., pp. 2935–2946.
- Pellegrini, Stefano, Ess, Andreas, Gool, Luc Van, 2010. Improving data association by joint modeling of pedestrian trajectories and groupings. In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 452–465.
- Robicquet, Alexandre, Sadeghian, Amir, Alahi, Alexandre, Savarese, Silvio, 2016. Learning social etiquette: Human trajectory understanding in crowded scenes. In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 549–565.
- Robin, Th, Antonini, Gianluca, Bierlaire, Michel, Cruz, Javier, 2009. Specification, estimation and validation of a pedestrian walking behavior model. *Transp. Res. B* 43 (1), 36–56.
- Sadeghian, Amir, Kosaraju, Vineet, Sadeghian, Ali, Hirose, Noriaki, Rezaatoghhi, Hamid, Savarese, Silvio, 2019. 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 (CVPR)*. pp. 1349–1358.
- Seer, Stefan, Brändle, Norbert, Ratti, Carlo, 2014. Kinects and human kinetics: A new approach for studying pedestrian behavior. *Transp. Res. C* 48, 212–228.
- Su, Jiawei, Vargas, Danilo Vasconcellos, Sakurai, Kouichi, 2019. One pixel attack for fooling deep neural networks. *IEEE Trans. Evol. Comput.* 23 (5), 828–841.
- Szegedy, Christian, Zaremba, Wojciech, Sutskever, Ilya, Bruna, Joan, Erhan, Dumitru, Goodfellow, Ian, Fergus, Rob, 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.
- Tsipras, Dimitris, Santurkar, Shibani, Engstrom, Logan, Turner, Alexander, Madry, Aleksander, 2018. Robustness may be at odds with accuracy. *arXiv Preprint arXiv:1805.12152*.
- van der Heiden, T., Nagaraja, N. S., Weiss, C., Gavves, E., 2019. Safecritic: Collision-aware trajectory prediction. In: *British Machine Vision Conference Workshop*.
- Vemula, Anirudh, Mueller, Katharina, Oh, Jean, 2018. Social attention: Modeling attention in human crowds. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1–7.
- Vizzari, Giuseppe, Manenti, Lorenza, Ohtsuka, Kazumichi, Shimura, Kenichiro, 2015. An agent-based pedestrian and group dynamics model applied to experimental and real-world scenarios. *J. Intell. Transp. Syst.* 19 (1), 32–45.
- Yu, Cunjun, Ma, Xiao, Ren, Jiawei, Zhao, Haiyu, Yi, Shuai, 2020. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 507–523.
- Zeng, Wenyan, Luo, Wenjie, Suo, Simon, Sadat, Abbas, Yang, Bin, Casas, Sergio, Urtasun, Raquel, 2019. End-to-end interpretable neural motion planner. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 8660–8669.
- Zhang, Wei Emma, Sheng, Quan Z., Alhazmi, Ahoud, Li, Chenliang, 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol. (TIST)* 11 (3), 1–41.