Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2017.DOI*

# CoL-GAN: Plausible and Collision-less Trajectory Prediction by Attention-based GAN

**SHAOHUA LIU[1,2], HAIBO LIU[1], HUIKUN BI[3], AND TIANLU MAO[3]**

[1]School of Electronic Engineering,Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Institute of Electronic and Information Engineering in Guangdong, University of Electronic Science and Technology of China, Dongguan, 523808, China
[3]Beijing Key Lab of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

Corresponding author: Tianlu Mao (ltm@ict.ac.cn)

This work is in part supported by the National Key Research and Development Program of China (2017YFC0804900), the National Natural Science Foundation of China (61532002), the Major Program of National Natural Science Foundation of China (91938301), the National Defense Equipment Advance Research Shared Technology Program of China (41402050301-170441402065), and the Sichuan Science and Technology Major Project on New Generation Artificial Intelligence (2018GZDZX0034).

**ABSTRACT** Predicting plausible and collisionless trajectories is critical in various applications, such as robotic navigation and autonomous driving. This is a challenging task due to two major factors. First, it is difficult for deep neural networks to understand how pedestrians move to avoid collisions and how they react to each other. Second, given observed trajectories, there are multiple possible and plausible trajectories followed by pedestrians. Although an increasing number of previous works have focused on modeling social interactions and multimodality, the trajectories generated by these methods still lead to many collisions. In this work, we propose CoL-GAN, a new attention-based generative adversarial network using a convolutional neural network as a discriminator, which is able to generate trajectories with fewer collisions. Through experimental comparisons with prior works on publicly available datasets, we demonstrate that Col-GAN achieves state-of-the-art performance in terms of accuracy and collision avoidance.

**INDEX TERMS** Trajectory prediction, generative adversarial network, deep learning.

## I. INTRODUCTION

Pedestrian trajectory prediction is essential and critical in various applications, such as autonomous driving and robotic navigation. When autonomous agents move in a crowd, they should understand social behaviors to avoid potential collisions with other agents. Social interactions and multimodality are two major inherent challenges of pedestrian trajectory prediction. There are several kinds of human-human interactions, such as avoiding collisions, grouping, and keeping a suitable distance with neighbors. Multimodality means that pedestrians will generate diverse trajectories, even if similar historical trajectories are given.

Recently, data-driven methods [1]–[8] have been proven to be able to capture more diverse and complicated social interactions than traditional methods [9], [10] that model social interactions by predefining physical rules. However, compared with data-driven methods, these traditional methods can generate collision-free trajectories. Indeed, pedestrians walking in crowds will try to avoid collisions with others, so being collision-free is a fundamental characteristic of real trajectories. In some scenarios of particular interest, produc-

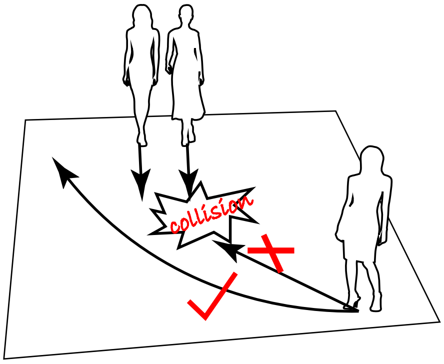


**FIGURE 1. Pedestrians walking in crowds will try to avoid collisions with others. So being collision-free is a fundamental characteristic of real trajectories.**

ing trajectories without collisions is necessary and critical. Maintaining a minimal distance, namely, a collision distance threshold, from others is necessary, even when pedestrians have to shorten the distance in an extremely crowded scenario. If autonomous driving systems predict trajectories with a large number of collisions, they will be likely to make wrong decisions. For example, if an autonomous vehicle forecasts that pedestrians will appear in the same location and move past them, which ignores the fact that they will occupy a larger area, accidents will be likely to happen.

Therefore, predicting collision-free trajectories is necessary. However, the existing data-driven trajectory prediction models [1], [4], [5], [8] suffer from a lack of quantitative evaluation metrics to evaluate the performance of collision avoidance and judge the plausibility of predicted trajectories. These models only use qualitative metrics, such as checking visualized trajectories in the prediction period, which is subjective, time-consuming, and not convincing enough. Here, we introduce the Average Collision Times (ACT), an intuitive quantitative metric to evaluate the plausibility of trajectories more convincingly.

As mentioned above, previous data-driven models predict trajectories with many collisions. Therefore, we propose an attention-based generative adversarial network (CoL-GAN), a novel data-driven model to predict pedestrian trajectories with fewer collisions. Experimental results show that Col-GAN can achieve higher accuracy and fewer collisions. The existing models utilize different schemes to model social behaviors. In contrast to prior works [2], [5], [7], [8] computing attention in complex ways, CoL-GAN adopts a novel social attention module with a simple structure to capture human-human interactions, where the attention scores are inferred based on relative positions and relative velocities. In contrast to works [1], [3], [5] considering the neighboring pedestrians in local areas, CoL-GAN focuses separately on all the pedestrians in a scenario. While Social GAN (SGAN) [4] uses its pooling mechanism only once to obtain social interaction features as the partial initial hidden state of the decoder, the attention module of CoL-GAN works at all time steps in the prediction period. In other words, our proposed attention module infers attention scores between the target pedestrian and all pedestrians (including the target pedestrian himself) based on the corresponding relative positions and relative velocities at each time step in the prediction period.

In addition to the complexity of social interactions between humans, another inherent property of trajectories is multimodality. Zhang *et al.* [11] used GMM (Gaussian mixture model) to estimate the probability distribution of a future position. However, a generative adversarial network can generate multimodality samples at the whole trajectory level. Similar to prior works [4], [6], [12], CoL-GAN also leverages a generative adversarial network architecture to produce multimodal trajectories. However, unlike previous LSTM-based discriminators, inspired by PatchGAN and PixelGAN [13], we introduce the Motion Discriminator, a CNN-based discriminator, which splits a whole trajectory into several parts to separately estimate the probability of being true.

Our contributions can be summarized as follows: (1) We introduce Col-GAN, a novel GAN model to predict human trajectories based on a new social attention mechanism. Col-GAN exploits a CNN-based network as the trajectory discriminator. (2) We introduce the ACT, a quantitative metric to evaluate the performance of collision avoidance, statistically and objectively. (3) Through experimental comparisons with state-of-the-art methods, in addition to the accuracy outperformance, the trajectories predicted by Col-GAN result in the best ACT.

To better present our work, the rest of this paper is arranged as follows. We describe related works in Section 2. Then, our method is introduced in detail in Section 3. We present our experimental results in Section 4. Finally, we conclude the paper in Section 5.

## II. RELATED WORK

### A. TRAJECTORY PREDICTION

Traditional works predicted trajectories by hand-crafted functions [9], [10], [14], [15]. Recently, more researchers have focused on solving this problem in a data-driven fashion, and many studies have made progress in this aspect. From the perspective of moving agents, these studies can be classified into heterogeneous agent trajectory prediction [16], [17] and homogeneous agent trajectory prediction [1]–[4], [6]. This work belongs to the latter. Depending on whether methods can predict multiple plausible future trajectories, they can be divided into deterministic methods [1]–[3], [5], [7], [18], [19] and stochastic methods [4], [6], [8], [12], [20].

Alahi *et al.* [1] proposed modeling human-human interactions by LSTM with a social pooling module, which led to the trend of data-driven methods. Because pedestrians pay different attention to others based on their motions and movements, more methods try to model this phenomenon by various attention modules [4]–[6], [8], [18]. The key differences among these attention modules involve three aspects. First, they compute the weights of pedestrians based on different source information. Second, they combine the motion mode of target pedestrians with interactive information of others in different ways. Third, they process human-human interactions at different frequencies.

### B. COLLISION AVOIDANCE

Being collision-free is the most intuitive phenomenon of plausible trajectories. Traditional trajectory prediction methods built their hand-crafted functions mainly inspired by modeling the collision avoidance phenomenon of pedestrians [9], [21], [22]. Such methods are strictly collision-free methods; at each time step, they predict trajectories with the purpose of avoiding collisions. In contrast, data-driven methods cannot predict trajectories without collisions. They implicitly learn to predict plausible trajectories from training data so that they can model more diverse interactions such as group forming and pedestrian following. S-LSTM [1] uses a pooling module to capture the interactions of pedestrians

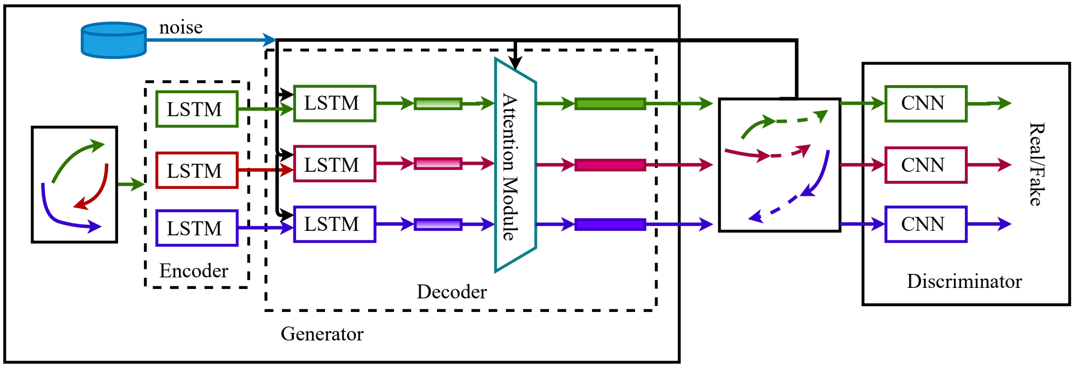


FIGURE 2. **An overview of CoL-GAN architecture.**

with the anticipation of helping predict trajectories with fewer collisions. Other attention-based methods [5]–[8], [18] also keep this in mind. Each method exhibits different performance in collision avoidance, and experimental results show that our method does especially well in avoiding collisions.

### C. GENERATIVE ADVERSARIAL NETWORK

The movements of pedestrians are multimodal in nature. Because generative adversarial networks (GANs) are able to generate multimodal samples, they are suitable for solving these problems. Gupta *et al.* [4] emphasized the problem of multimodality of trajectories and introduced SGAN. Later, other works [6], [8], [12], [12] also tried to generate more diverse trajectories. Great advances have been made in GANs in many fields. A GAN is composed of a generator and a discriminator. Many works have focused on designing task-fit generators to make improvements [13], [23]–[25]. Additionally, other methods have tried to make progress by inventing effective discriminators [13], [26]. The discriminator used in PatchGAN and PixelGAN [13] has been proven to be more useful in generating more realistic pictures than a traditional picture discriminator. Inspired by these works, we use a CNN-based motion discriminator rather than a traditional LSTM-based trajectory discriminator.

## III. METHOD

Pedestrians walking in a crowd perform diverse social interactions with others at every time step. They respond differently to others depending on the relative positions and relative velocities between them. They keep pace with nearby pedestrians who have the same destinations while carefully maintaining a proper distance with them so as not to cause a collision. However, different pedestrians prefer to maintain different proper distances. If there are pedestrians walking toward them, they may preferentially adjust their directions and speeds in advance to avoid collisions. However, they will choose different avoidance directions and speeds. In other words, pedestrians pay different attention to others in various scenarios, and there are a variety of ways for them to choose to avoid collisions with others.

This motivates us to build a model that can integrate the aforementioned characteristics to predict multimodal future trajectories with fewer collisions. Therefore, we present a seq2seq-based GAN model to predict diverse trajectories, with an attention module to deal with interactions at each prediction time step, and this model can compute different attention scores of pedestrians based on their relative positions and relative velocities to the target pedestrian. We call the model collisionless GAN, CoL-GAN. The architecture of CoL-GAN is presented in Fig. 2.

### A. PROBLEM DEFINITION

In this paper, we address the problem of predicting accurate pedestrian trajectories with fewer collisions in crowded scenarios. Given the historical trajectories of all pedestrians in the scenario, our task is to predict their future trajectories simultaneously. The pedestrians in the scenario are represented as $p_1, p_2, ..., p_N$. The position of a specific pedestrian $p_i(i \in [1, N])$ at any historical time step $t(t \in [1, T_{obs}])$ is defined as $X_i^t = (x_i^t, y_i^t)$. Our goal is to predict the positions of pedestrians at any future time step $t(t \in [T_{obs}+1, T_{obs}+T_{pred}])$, and for a specific pedestrian $p_i(i \in [1, N])$, the predicted position will be $\hat{Y}_i^t = (\hat{x}_i^t, \hat{y}_i^t)$, while the ground truth is defined as $Y_i^t = (x_i^t, y_i^t)$.

### B. OVERALL MODEL

CoL-GAN is a typical generative adversarial network composed of a generator and a discriminator. The generator generates future trajectories of pedestrians, and the discriminator estimates the probability that they are true.

Our generator is based on a seq2seq architecture with a historical trajectory encoder and a future trajectory decoder. The key difference between the LSTM-based encoder and decoder is that the decoder has a social attention module. The encoder is used to encode historical trajectories. It captures each pedestrian's historical motion patterns and encodes them into the hidden state $h_{en}^t$ and the cell state $c_{en}^t$. At the time step $t_{obs}$, the encoder provides $h_{en}^{t_{obs}}$ and $c_{en}^{t_{obs}}$ to the LSTM of our decoder as the initial hidden state and cell state, respectively. The decoder is the key component of our design; it consists of three components, an LSTM ($LSTM_{de}$), a social attention module, and a linear layer. Pedestrians walking in a crowd avoid collisions with others at every time step. They observe the movements and velocities of others and then plan their own routes. Finally, they walk following their planned paths with some urgent collision avoidance. Therefore, our decoder also uses the social attention module to process interactions at each prediction time step. For the target pedestrian $p_i$, the attention module uses an MLP to infer all pedestrians' corresponding weights by using their relative positions and relative velocities to $p_i$ and then computes a weighted sum of the corresponding outputs of $LSTM_{de}$. Finally, the weighted sum is used by the linear layer to predict future movements. In addition, noise and the predictions of the last time step will be concatenated as the input of $LSTM_{de}$. For different trajectories, the noise will be different, but it is time-invariant.

After the generator predicts all trajectories, our CNN-based discriminator determines whether they are true or false to force the generator to predict more realistic trajectories. In the following subsections, we elaborate on each module in detail.

### C. TRAJECTORY ENCODER

From historical trajectories of pedestrians, the encoder will capture their respective motion patterns. We do not directly use the coordinate $X_i^t = (x_i^t, y_i^t)$ as the input of the encoder. Following the process of SGAN [4] and STGAT [8], we also use $\Delta X_i^t = (\Delta x_i^t, \Delta y_i^t)$ as the input, which is equivalent to the velocity. Their definitions are as follows:

$$\begin{aligned} \Delta X_i^t &= X_i^t - X_i^{t-1} \\ \Delta x_i^t &= x_i^t - x_i^{t-1} \\ \Delta y_i^t &= y_i^t - y_i^{t-1} \end{aligned} \tag{1}$$

Then $\Delta X_i^t$ is embedded into a fixed-length vector $e_{(en,i)}^t$ at every time-step. $LSTM_{en}$ will use $e_{(en,i)}^t$ as the input and produce a new hidden state $h_{(en,i)}^t$ as follows:

$$\begin{aligned} e_{(en,i)}^t &= \phi_{en}\left(\Delta X_i^t; W_{(en,em)}\right) \\ h_{(en,i)}^t &= LSTM_{en}\left(h_{(en,i)}^{t-1}, e_{(en,i)}^t; W_{(en,lstm)}\right) \end{aligned} \tag{2}$$

where $\phi_{en}(\cdot)$ is an embedding function and $W_{(en,em)}$ are embedding weights. The LSTM weights of the encoder are denoted by $W_{(en,lstm)}$.

### D. ATTENTION BASED TRAJECTORY DECODER

At every time step $t$, our decoder takes the $\Delta\hat{Y}_i^{t-1}$ predicted at last time step as its input. Specifically, at the first prediction time step $T_{obs}+1$, the input is $\Delta Y_i^{T_{obs}}$, and $LSTM_{de}$ takes $[h_{(en,i)}^{T_{obs}}, c_{(en,i)}^{T_{obs}}]$ as the initial hidden state and cell state, respectively. Similar to the encoder, $\Delta\hat{Y}_i^t$ is also embedded into a fixed-length vector $e_{(de,i)}^t$ at every time-step. Then, $e_{(de,i)}^t$ and the noise vector $z$ are concatenated together and are fed into the LSTM of the decoder. Therefore, this introduces the following recursion:

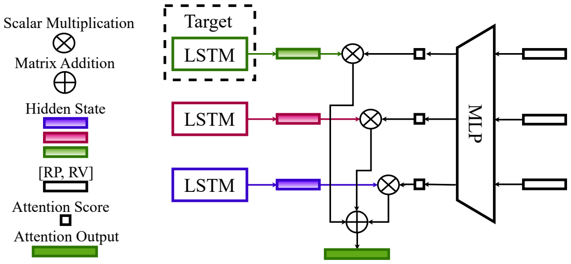


**FIGURE 3.** Illustration of attention module architecture. It assigns different attention scores to pedestrians including the target pedestrian.

$$\begin{aligned} e_{(de,i)}^t &= \phi_{de}\left(\Delta\hat{Y}_i^t; W_{(de,em)}\right) \\ h_{(de,i)}^t &= LSTM_{de}\left(h_{(de,i)}^{t-1}, e_{(de,i)}^t, z; W_{(de,lstm)}\right) \end{aligned} \tag{3}$$

where $\phi_{de}(\cdot)$ is the embedding function of the decoder and $W_{(de,em)}$ are its embedding weights. The LSTM weights of the decoder are denoted by $W_{(de,lstm)}$. Here, a noise vector $z$ is sampled from a Gaussian distribution, which is person-specific but time-invariant.

**Social Attention Module**. At every time step, pedestrians consider the movements of others and make a plan to avoid collisions with others while walking toward their own destinations. The main factors that influence their decisions are the relative positions and relative velocities between them and others. Clearly, the naive LSTM is not able to capture interactions between pedestrians, but our social attention module does. Unlike S-LSTM [1] which limits the local area in which interactions are considered, our attention module considers all pedestrians in the scenario. Unlike SGAN [4] and STGAT [8], which capture social interaction features to initialize the hidden state of the decoder, our attention module works throughout the prediction period, handling social interaction at each time step. While CIDNN [19] only uses absolute positions to infer attention weights, we take into account both the positions and velocities. We denote the relative positions from pedestrian $p_i$ to target $p_j$ as $RP_{ij}$ and the relative velocities as $RV_{ij}$. $RP_{ij}$ and $RV_{ij}$ are calculated as follows:

$$\begin{aligned} RP_{ij} &= \hat{Y}_i - \hat{Y}_j \\ RV_{ij} &= \Delta\hat{Y}_i - \Delta\hat{Y}_j \end{aligned} \tag{4}$$

Then, $RP_{ij}$ and $RV_{ij}$ are concatenated as $R_{ij}$, which is the input of an MLP. For target pedestrian $p_j$, every pedestrian in the scenario (including $p_j$) will be assigned an attention score $\alpha_{ij}^t$ calculated by the MLP with a softmax layer to normalize the probabilities at time step $t$. $H_j^t$ of target $p_j$ will be calculated as the weighted sum of $h_i^t (i \in [1, N])$ weighted by $\alpha_{ij}^t$. The formulations are as follows:

$$\alpha_{ij}^t = \text{softmax}\left(MLP\left(R_{ij}^t; W_{mlp}\right)\right) \tag{5}$$

$$H_j^t = \sum_{i \in [1,\ldots,N]} h_i^t \alpha_{ij}^t \tag{6}$$

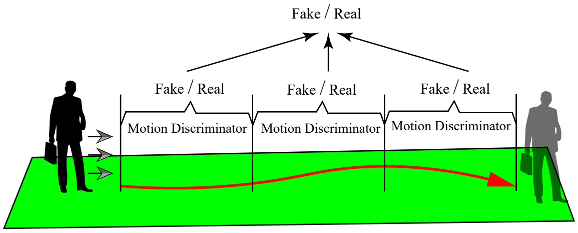


**FIGURE 4.** **Illustration of how Motion Discriminator works. Our Motion Discriminator classifies whether each segment of a trajectory is real or fake.**

Then $H_j^t$ will be used to predict $\Delta\hat{Y}_j^{t+1}$ as follows:

$$\Delta\hat{Y}_j^{t+1} = linear\left(H_j^t; W_l\right) \tag{7}$$

### E. CNN-BASED MOTION DISCRIMINATOR

We are inspired by PatchGAN and PixelGAN [13], which do not classify whether a whole image is real or fake but classify whether each $N \times N$ patch in an image is real or fake. Therefore, we abandoned the typical discriminators classifying whether a whole trajectory is real or fake in SGAN [4] and SoPhie [6]. Our Motion Discriminator classifies whether each segment of a trajectory is real or fake. Because LSTMs may suffer from vanishing gradients, partly neglecting previous information, we use a fully convolutional network to work as the discriminator. As mentioned before, our generator generates $[\Delta\hat{Y}_i^{T_{obs}+1}, \Delta\hat{Y}_i^{T_{obs}+2}, \Delta\hat{Y}_i^{T_{obs}+T_{pred}}]$ which is the sequence of velocities of a pedestrian. We assume that the movements of pedestrians at any time step are equally important. Moreover, pedestrians can make correct decisions and move reasonably at every time step. Therefore, the probability of a trajectory being real is the average probability of all its segments. Our discriminator has three 1-D convolutional layers. Their kernel sizes are all set to 1. The convolution stride is fixed to 1, and the padding is set to 0. For the first two layers, LeakyReLU is used as our non-linear activation function. Between the second convolutional layer and the last convolutional layer, a batch normalization technique is used. Finally, a sigmoid function is used to compute the probability. After Motion Discriminator obtains the probability of each segment, we average them as the whole probability of the trajectory.

### F. IMPLEMENTATION AND TRAINING DETAILS

We use an embedding dimension of 32, and the LSTM hidden state dimensions of both the encoder and decoder are 64. All LSTMs have 1 layer. The dimensions of the MLP are $[16, 32, 1]$. When training the model, we use both the original GAN loss and the variety loss proposed by SGAN [4]. However, for the variety loss, we use L1 instead of L2. For consideration of the training time, we do not set hyperparameter $k$ of the variety loss to be 20 like SGAN [4] and STGAT [8], but we set it to only 5. We iteratively train the generator and the discriminator with a batch size of 32 for 200 epochs using Adam with an initial generator learning rate of 0.001 and a discriminator learning rate of 0.00001. After 20 epochs, the generator learning rate is set to 0.0001. Our implementation is based on the PyTorch library. The model is trained on one Nvidia GeForce GTX 1080 Ti graphics card.

## IV. EXPERIMENTS

### A. DATASETS AND METRICS

In this section, we evaluate our method on two public pedestrian-trajectory datasets: ETH [27] and UCY [28]. The ETH dataset consists of two scenarios named ETH and HOTEL. UCY includes two scenarios that are divided into 3 parts, named ZARA1, ZARA2 and UNIV. Among these five subdatasets, UNIV is the most crowded. We follow the same data preprocessing strategy as SGAN [4] and STGAT [8], and there are no other processes. All data are converted to a world coordinate system and then interpolated to obtain values every 0.4 s. We observe 8 time steps ($T_{obs} = 8$) of trajectories to predict the following 12 time steps ($T_{pred} = 12$). In addition, we follow the leave-one-out evaluation methodology in SGAN [4], training on 4 subdatasets and testing on the remaining subdataset.

**Quantitative Evaluation Metrics**. Two metrics were widely used by previous works to evaluate the performance of trajectory prediction, the Average Displacement Error (ADE) and the Final Displacement Error (FDE). **ADE**: The average Euclidean distance between the ground truth and the predicted trajectories of all predicted time steps. **FDE**: Euclidean distance between the ground truth and the predicted position at the final time step ($T_{obs} + T_{pred}$). The ADE and FDE can be used to quantitatively evaluate the accuracy of predicted trajectories, while there are no quantitative evaluation metrics to evaluate the reasonability and sociality of predicted trajectories. Since being collision-free is the most significant representation of human-human interactions [9], [10], we introduce the **Average Collision Times (ACT)** to quantitatively evaluate the reasonability of trajectories. It can be calculated as follows:

$$ACT = \frac{\sum_{m=1}^{M} \sum_{t=T_{obs}+1}^{T_{obs}+T_{pred}} C_m^t}{M} \tag{8}$$

where $C_m^t$ represents the collision times in a scene at frame $t$ and $M$ is the number of scenes. At time step $t$, if the distance between pedestrian $p_i$ and pedestrian $p_j$ is less than a threshold $D_{thr}$, there is a collision, and for $p_i$ and $p_j$, the collision should be counted only once. In our experiments, $D_{thr}$ is set to 0.3 m and we also compare the performance of different methods for other different $D_{thr}$ values from 0.08 m to 0.35 m.

**Baselines**. We compare against the following baselines:

- **LSTM**: A vanilla LSTM without an interaction mechanism.
- **S-LSTM**: The method proposed by Alahi *et al.* [1]. Each pedestrian is modeled with an LSTM, and the hidden state is pooled with neighbors at each time-step.

**TABLE 1.** ADE&FDE results of all methods across datasets. Because the original video of ETH is an accelerated version that is mentioned in SRLSTM [5], we also use the frame rate-corrected version of ETH, ETH-SR, which is also used by SRLSTM [5]. All methods observe the trajectories of 8 time steps to predict the next 12 time steps. We draw 20 samples for all stochastic methods. We obtained the results of SGAN and SGAN-P by evaluating their own trained models released on Github.

| Metric | Dataset | Deterministic Models | | | Stochastic Models | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | LSTM | S-LSTM | CIDNN | SoPhie | SGAN | SGAN-P | STGAT | CoL-GAN(ours) |
| ADE/FDE | ETH | 1.09/2.41 | 1.09/2.35 | 1.25/2.32 | 0.70/1.43 | 0.70/1.29 | 0.77/1.40 | **0.65/1.12** | 0.78/1.52 |
| | ETH-SR | -/- | -/- | -/- | -/- | 0.56/1.08 | 0.55/1.05 | 0.56/1.08 | **0.48/0.93** |
| | HOTEL | 0.86 /1.91 | 0.79/1.76 | 1.31/ 2.36 | 0.76/1.67 | 0.48/1.02 | 0.43/0.87 | 0.35/0.66 | **0.27/0.46** |
| | UNIV | 0.61/1.31 | 0.67/1.40 | 0.90/1.86 | 0.54/1.24 | 0.56/1.18 | 0.75/1.50 | **0.52/1.10** | 0.53/1.12 |
| | ZARA1 | 0.41/0.88 | 0.47/1.00 | 0.50/1.04 | **0.30/0.63** | 0.34/0.68 | 0.35/0.69 | 0.34/0.69 | 0.33/0.68 |
| | ZARA2 | 0.52/1.11 | 0.56/1.17 | 0.51/1.07 | 0.38/0.78 | 0.31/0.65 | 0.36/0.72 | 0.29/0.60 | **0.27/0.58** |

**TABLE 2.** ACT results of all stochastic methods across datasets. The distance threshold of collisions is set to 0.3 m. ACT-best means we select the best scenario for the whole prediction duration of 20 samples of stochastic methods, which is similar to ADE and FDE. ACT-avg means we average the ACTs of 20 samples.

| Metric | Dataset | SGAN | SGAN-P | STGAT | CoL-GAN(ours) |
|---|---|---|---|---|---|
| ACT-best<br>($D_{thr} = 0.3m$) | ETH | 0.0857 | 0.0429 | 0.3000 | **0** |
| | ETH-SR | 0.0995 | 0.0962 | 0.2570 | **0.0033** |
| | HOTEL | 0.1667 | 0.1561 | 0.1063 | **0.0233** |
| | UNIV | 7.4287 | 9.3073 | 7.9314 | **6.5375** |
| | ZARA1 | 0.0199 | 0.0499 | 0.0764 | **0.0083** |
| | ZARA2 | 0.4441 | 1.9045 | 0.5679 | **0.2780** |
| ACT-avg<br>($D_{thr} = 0.3m$) | ETH | 0.3436 | 0.3914 | 0.4207 | **0.1743** |
| | ETH-SR | 0.4090 | 0.4823 | 0.5038 | **0.1895** |
| | HOTEL | 0.4274 | 0.4650 | 0.3213 | **0.1015** |
| | UNIV | 12.2721 | 16.6796 | 12.3219 | **10.9481** |
| | ZARA1 | **0.2021** | 0.4390 | 0.3389 | 0.2435 |
| | ZARA2 | 1.1583 | 3.8390 | 1.1950 | **0.7389** |

- **CIDNN**: A method encoding crowd interaction with deep neural network [19].
- **SGAN**: The first method using a GAN to deal with the multi-modality of trajectories [4].
- **SoPhie**: A GAN-based method that leverages both social and physical information [6].
- **STGAT**: A method that can model spatial-temporal interactions for pedestrian trajectory prediction by using graph attention networks [8].

## B. QUANTITATIVE EVALUATION

We compare our method on three metrics ADE, FDE and ACT ($D_{thr}$=0.3 m) against different baselines in Table 1 and Table 2. Fig. 5 and Fig. 6 show the collision avoidance performance of different methods for different $D_{thr}$ values from 0.08 m to 0.35 m. When computing the ADE and FDE, for stochastic models, we draw 20 samples following SGAN [4] and STGAT [8], and we used the same evaluation codes as them. For the ACT results, we compare two types, ACT-best and ACT-avg. The former means we select the best sample, and the latter means we average the ACTs of 20 samples.

**ADE & FDE**. For the ADE and FDE, Table 1 shows the superior performance of different models on different subdatasets. SoPhie outperforms others on ZARA1. STGAT obtains the lowest ADE and FDE on ETH and UNIV. Our model makes great progress on ETH-SR, HOTEL and ZARA2. On the whole, CoL-GAN obtains comparable results with other state-of-the-art methods and outperforms them on several specific subdatasets. As shown in Table 1, SGAN is better than SGAN-P, and it seems that its social pooling module does not help to improve the prediction accuracy. In contrast, STGAT and CoL-GAN show superiority in accuracy. Our model achieves poor performance on ETH but achieves the best performance on ETH-SR. This is mainly because the original video of ETH is an accelerated version that is mentioned in SRLSTM [5], while ETH-SR is the frame rate-corrected ETH. Our Motion Discriminator seems to be sensitive to the unusual $\Delta\hat{Y}_i^t$.

**ACT**. As shown in Table 2, CoL-GAN exhibits excellent collision avoidance performance as evaluated by both ACT-best and ACT-avg. For ACT-best, CoL-GAN outperforms the other methods on all subdatasets. For ACT-avg, CoL-GAN outperforms the other methods on nearly all subdatasets except ZARA1. While ACT-best presents the upper-limit performance of stochastic models, ACT-avg shows the average performance. As shown in Table 2, for ACT-avg, SGAN-P is once again worse than SGAN on all subdatasets. Compared with SGAN, STGAT does not perform better in collision avoidance, while CoL-GAN performs better than SGAN. To better compare the collision avoidance performance of different stochastic methods, we plot Fig. 5 and Fig. 6 to show the ACT-best and ACT-avg values for different distance thresholds. As Fig. 5 and Fig. 6 show, CoL-GAN stably outperforms the other methods.

### 1) Ablation Study

We elaborate on an ablation study to confirm the effectiveness of our social attention module and Motion Discriminator. We present three variations of CoL-GAN: CoL-GAN without a social module (CoL-GAN-noAttn), CoL-GAN with the discriminator of SGAN (CoL-GAN-tradD), and CoL-GAN with a self-attention module [25] (CoL-GAN-selfAttn). CoL-GAN-selfAttn is a variation in which the attention module is replaced with the self-attention module presented in Self-Attention-GAN [25]. The self-attention module computes the attention score $\alpha$ by measuring the intrinsic similarity of LSTM hidden states of different pedestrians. We present SGAN-ourD, an SGAN trained with our Motion Discriminator. Table 3 demonstrates that CoL-GAN performs better than all the other methods in terms of ADE, FDE, ACT-best and ACT-avg on most of the subdatasets.

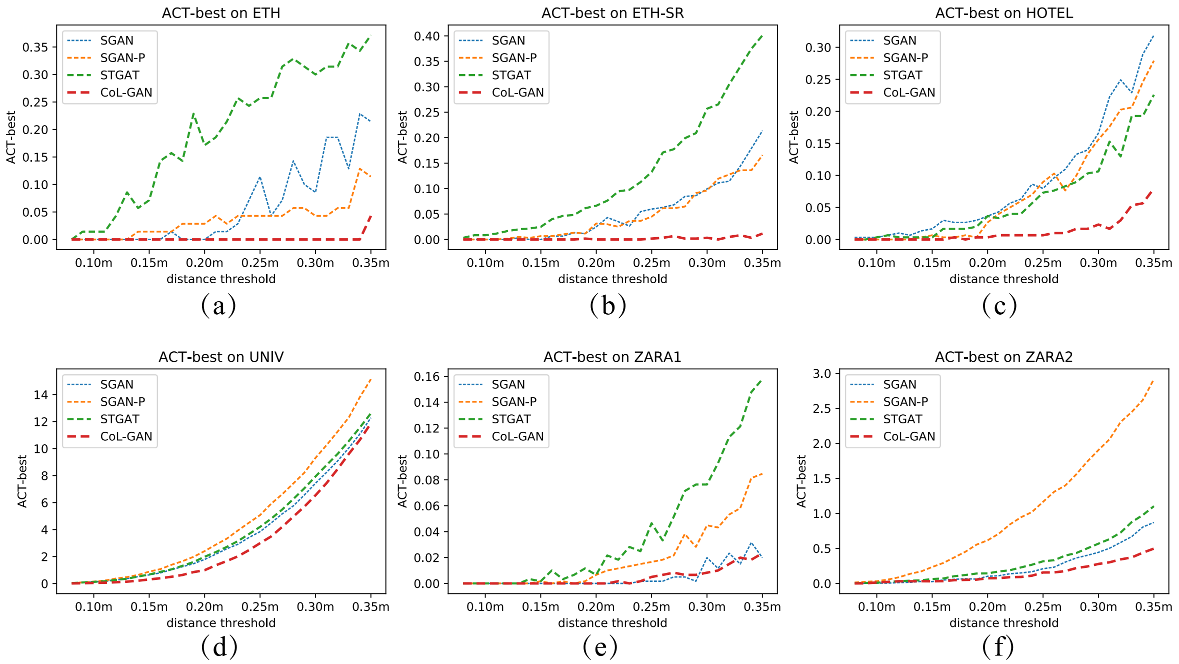


**FIGURE 5.** The upper limit collision avoidance performance of different methods evaluated for different distance thresholds from 0.08 m to 0.35 m on different datasets. (a) ACT-best on ETH. (b) ACT-best on ETH-SR. (c) ACT-best on HOTEL. (d) ACT-best on UNIV. (e) ACT-best on ZARA1. (f) ACT-best on ZARA2.

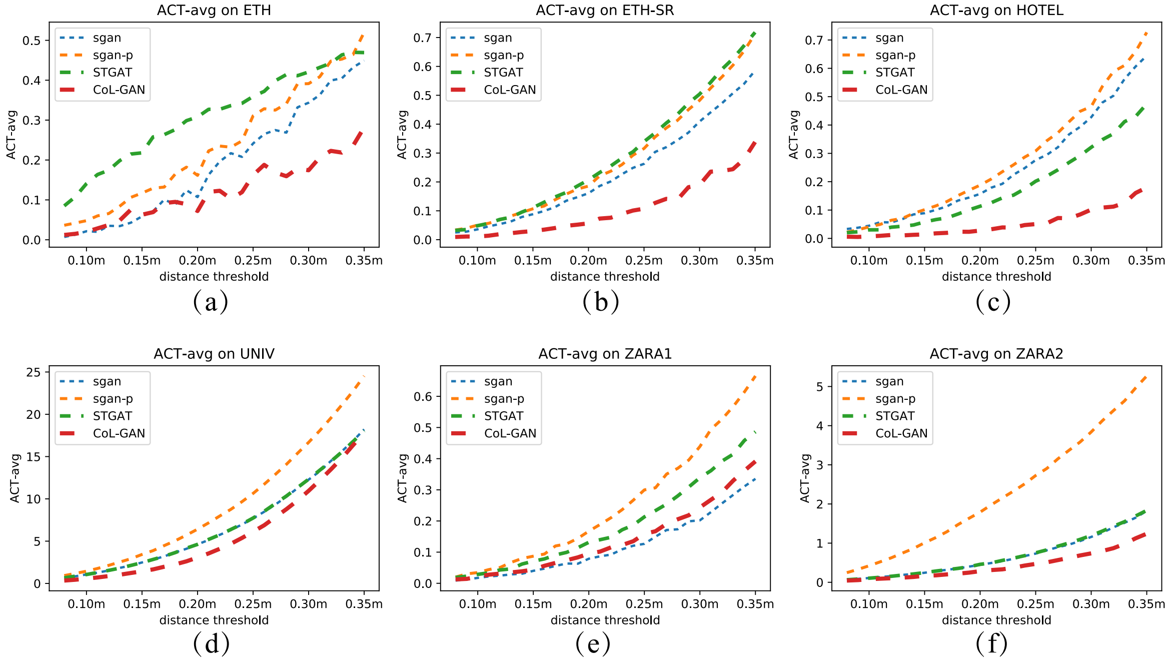


**FIGURE 6.** The average collision avoidance performance of different methods evaluated for different distance thresholds from 0.08 m to 0.35 m on different datasets. (a) ACT-avg on ETH. (b) ACT-avg on ETH-SR. (c) ACT-avg on HOTEL. (d) ACT-avg on UNIV. (e) ACT-avg on ZARA1. (f) ACT-avg on ZARA2.

**SGAN vs SGAN-ourD and CoL-GAN-tradD vs CoL-GAN.** As Table 3 shows, for the ADE and FDE, SGAN-ourD achieves a considerable improvement over SGAN on nearly all subdatasets and CoL-GAN outperforms CoL-GAN-tradD, which indicates the superiority of the Motion Discriminator. Compared with other subdatasets, on HOTEL, our discriminator helps improve accuracy the most.

**TABLE 3. Experiments to validate the superiority of our Motion Discriminator and attention module. For SGAN-ourD, we only replace its discriminator with ours. CoL-GAN-tradD is CoL-GAN trained with a traditional LSTM-based discriminator. CoL-GAN-noAttn is the CoL-GAN model without an attention module. CoL-GAN-selfAttn is the CoL-GAN model with a self-attention module [25] instead of our attention module.**

| Metric | Dataset | SGAN | SGAN-ourD | CoL-GAN-selfAttn | CoL-GAN-noAttn | CoL-GAN-tradD | CoL-GAN |
|---|---|---|---|---|---|---|---|
| ADE/FDE | ETH-SR | 0.56/1.08 | 0.55/1.06 | 0.53/1.00 | 0.54/1.07 | 0.49/0.96 | **0.48/0.93** |
| | HOTEL | 0.48/1.02 | 0.30/0.55 | 0.31/0.58 | 0.30/0.55 | 0.32/0.59 | **0.27/0.46** |
| | UNIV | 0.56/1.18 | 0.52/1.11 | 0.51/1.10 | **0.50/1.09** | 0.54/1.14 | 0.53/1.12 |
| | ZARA1 | 0.34/0.68 | 0.35/0.72 | 0.34/0.70 | 0.34/0.73 | **0.32/0.66** | 0.33/0.68 |
| | ZARA2 | 0.31/0.65 | 0.28/0.58 | 0.28/0.60 | 0.28/0.60 | 0.30/0.65 | **0.27/0.58** |
| ACT-best ($D_{thr} = 0.3m$) | ETH-SR | 0.0995 | 0.0464 | 0.0199 | 0.0133 | 0.0083 | **0.0033** |
| | HOTEL | 0.1667 | 0.0930 | 0.1063 | 0.0963 | **0.0100** | 0.0233 |
| | UNIV | 7.4287 | 6.8057 | **5.9345** | 6.1457 | 6.6938 | 6.5375 |
| | ZARA1 | 0.0199 | 0.0465 | 0.0282 | 0.0233 | 0.0133 | **0.0083** |
| | ZARA2 | 0.4441 | 0.4213 | 0.2888 | 0.2400 | **0.1564** | 0.2780 |
| ACT-avg ($D_{thr} = 0.3m$) | ETH-SR | 0.4090 | 0.4993 | 0.6476 | 0.4344 | 0.4456 | **0.1895** |
| | HOTEL | 0.4274 | 0.3920 | 0.4189 | 0.3985 | 0.4229 | **0.1015** |
| | UNIV | 12.2721 | 12.4466 | 12.9163 | 12.7872 | 11.5638 | **10.9481** |
| | ZARA1 | 0.2021 | 0.4094 | 0.4909 | 0.5597 | **0.1309** | 0.2435 |
| | ZARA2 | 1.1583 | 1.2571 | 1.3229 | 1.2562 | 0.9626 | **0.7389** |

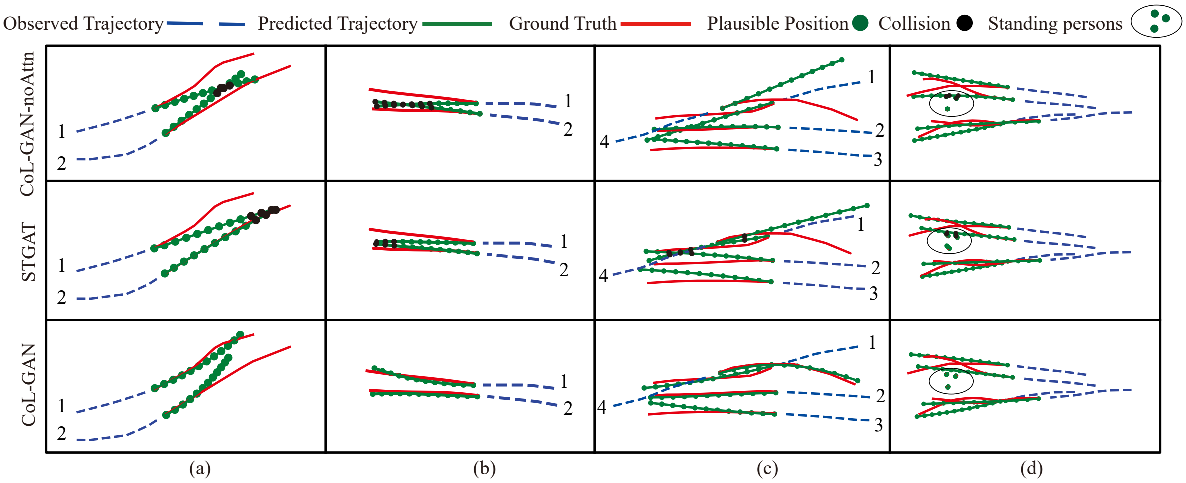


**FIGURE 7. Comparisons of our model with STGAT and our variation model without the interaction module (CoL-GAN-noAttn) in four different scenarios. To highlight the performances of different models, only a portion of the pedestrians in the scene are presented. It is obvious that our attention module boosts the performance of our model to generate more plausible trajectories.**

**CoL-GAN vs CoL-GAN-noAttn and CoL-GAN-selfAttn.** Compared with CoL-GAN-selfAttn and CoL-GAN-noAttn, as shown in Table 3, CoL-GAN has better ADE, FDE and ACT-best values on most datasets. For ACT-avg, CoL-GAN outperforms the other two methods on every subdataset.

### C. QUALITATIVE EVALUATION

The qualitative results are shown in Fig. 7. We choose several different scenarios. In Fig. 7(a), the prediction of CoL-GAN shows pedestrian #2 slows down his (her) speed to avoid collisions with pedestrian #1 while collisions happen in the results of SGAN and STGAT. As shown in Fig. 7(b), the trajectories predicted by CoL-GAN are closer to the ground truth, and they maintain the relative position relationship of pedestrians, which successfully avoids collisions. As shown in Fig. 7(c), CoL-GAN-noAttn predicts much more linear results than STGAT and CoL-GAN. In the results of STGAT, when pedestrian #1 and pedestrian #4 try to avoid each other, pedestrian #1 causes a collision with pedestrian #2. However, CoL-GAN predicts the most accurate trajectories without collisions. As shown in Fig. 7(d), for the results of STGAT and CoL-GAN-no-Attn, pedestrians cause collisions with other pedestrians standing there, while this does not occur in the results of our method.

Fig. 8 shows visualization examples of the learned attention weights in the social attention module. As shown in Fig. 8(a)-(d), CoL-GAN assigns different attention weights to surrounding pedestrians. The target pedestrians have the most importance. The relative importance of others depends on the distance and the similarity of velocity. As shown in Fig. 8(b), with an increase in the distance from the target pedestrian at two different time steps, the attention weight of the same pedestrian decreases.

## V. CONCLUSION

In this work, we propose an attention-based GAN with a CNN-based discriminator to predict pedestrian trajectories.

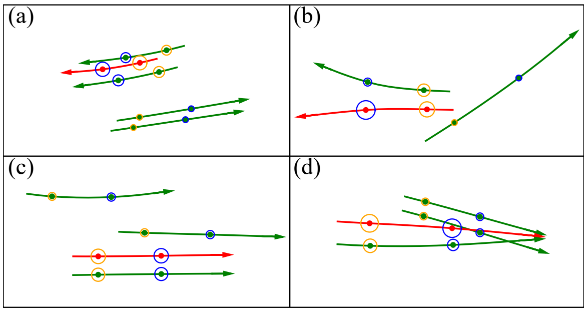


**FIGURE 8. The learned attention weights in the social attention module. Solid dots on a trajectory indicate different time steps and arrows show the directions of trajectories. The red trajectories are the target pedestrians and the green trajectories are the neighbors. Circles on trajectories show the attention represented by the radius proportional to the attention weight and circles with the same color indicate that they are at the same time step.**

Experimental results demonstrate that CoL-GAN is able to predict trajectories with higher accuracy and fewer collisions. Our attention module assigns different weights to the corresponding pedestrians including the target pedestrians to fuse human-human interactive information with the motion patterns of target pedestrians. Our Motion Discriminator classifies whether a trajectory is fake or real by classifying whether each segment of the trajectory is fake or real. To quantitatively evaluate the collision avoidance performance of data-driven methods, we introduce a new metric, the ACT. Experimental results for the ADE, FDE, and ACT demonstrate that our model outperforms other methods in terms of accuracy and collision avoidance. Our attention module shows state-of-the-art effects on avoiding collisions; furthermore, our discriminator helps the generator learn to predict more accurate results.

## REFERENCES

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 961–971.

[2] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 1–7.

[3] I. Hasan, F. Setti, T. Tsesmelis, A. Del Bue, F. Galasso, and M. Cristani, "Mx-lstm: mixing tracklets and vislets to jointly forecast trajectories and head poses," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6067–6076.

[4] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2255–2264.

[5] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, "Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 12 085–12 094.

[6] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1349–1358.

[7] J. Liang, L. Jiang, J. C. Niebles, A. G. Hauptmann, and L. Fei-Fei, "Peeking into the future: Predicting future person activities and locations in videos," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5725–5734.

[8] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, "Stgat: Modeling spatial-temporal interactions for human trajectory prediction," in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6272–6281.

[9] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," Physical review E, vol. 51, no. 5, p. 4282, 1995.

[10] Y. Ma, D. Manocha, and W. Wang, "Autorvo: Local navigation with dynamic constraints in dense heterogeneous traffic," arXiv preprint arXiv:1804.02915, 2018.

[11] W. Zhang, L. Sun, X. Wang, Z. Huang, and B. Li, "Seabig: A deep learning based method for location prediction in pedestrian semantic trajectories," IEEE Access, vol. PP, pp. 1–1, 08 2019.

[12] Y. Li, "Which way are you going? imitative decision learning for path forecasting in dynamic scenes," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 294–303.

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1125–1134.

[14] M. K. C. Tay and C. Laugier, "Modelling smooth paths using gaussian processes," in Field and Service Robotics. Springer, 2008, pp. 381–390.

[15] L. Tian and C. Collins, "An effective robot trajectory planning method using a genetic algorithm," Mechatronics, vol. 14, no. 5, pp. 455–470, 2004.

[16] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "Trafficpredict: Trajectory prediction for heterogeneous traffic-agents," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 6120–6127.

[17] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, "Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 8483–8492.

[18] A. Sadeghian, F. Legros, M. Voisin, R. Vesel, A. Alahi, and S. Savarese, "Car-net: Clairvoyant attentive recurrent network," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 151–167.

[19] Y. Xu, Z. Piao, and S. Gao, "Encoding crowd interaction with deep neural network for pedestrian trajectory prediction," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5275–5284.

[20] J. Amirian, J.-B. Hayet, and J. Pettré, "Social ways: Learning multi-modal distributions of pedestrian trajectories with gans," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019, pp. 0–0.

[21] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301), vol. 3. IEEE, 2002, pp. 1936–1941.

[22] R. W. Cox, "Social forces, states and world orders: beyond international relations theory," Millennium, vol. 10, no. 2, pp. 126–155, 1981.

[23] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "pix2pixhd: High-resolution image synthesis and semantic manipulation with conditional gans."

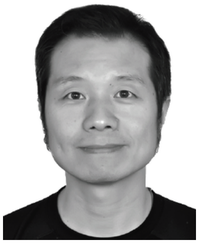
[24] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," arXiv preprint arXiv:1710.10196, 2017.

[25] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," arXiv preprint arXiv:1805.08318, 2018.

[26] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," arXiv preprint arXiv:1808.06601, 2018.

[27] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in 2009 IEEE 12th International Conference on Computer Vision. IEEE, 2009, pp. 261–268.

[28] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in Computer graphics forum, vol. 26, no. 3. Wiley Online Library, 2007, pp. 655–664.



SHAOHUA LIU received his B.S. and M.Sc. degrees from Zhejiang University in 1998 and 2001, respectively. Then, he received his Ph.D. in computer science from Institute of Software, Chinese Academy of Sciences in 2006. He is now an associate professor at the School of Electronic Engineering, Beijing University of Posts and Telecommunications, China. His major interests involve telecommunications engineering and software engineering.

HAIBO LIU is currently pursuing the M.Sc. degree. From 2018, he is studying in the School of Electronic Engineering, Beijing University of Posts and Telecommunications. Beijing, China. His research interests include deep learning and computer vision.

HUIKUN BI is an assistant professor with Institute of Computing Technology, Chinese Academy of Sciences. She received her Ph.D. degree in Computer Science from the Institute of Computing Technology, Chinese Academy of Sciences and University of Chinese Academy of Sciences. Her research interests include crowd simulation and human behavior prediction.

TIANLU MAO received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2009, where she is also working as associate professor. Her research interests include computer graphics and computer vision.