SCSG Attention: A Self-centered Star Graph with Attention for Pedestrian Trajectory Prediction

Xu Chen[1], Shuncheng Liu[1], Zhi Xu[1], Yupeng Diao[1], Shaozhi Wu[2], Kai Zheng[1,2], and Han Su[1,2(✉)]

[1] School of Computer Science and Engineering, Chengdu, China
{xuchen,liushuncheng,zhixu023,yupengdiao}@std.uestc.edu.cn,
{zhengkai,hansu}@uestc.edu.cn
[2] Yangtze Delta Region Institute (Quzhou),
University of Electronic Science and Technology of China, Chengdu, China
wszfrank@uestc.edu.cn

Abstract. Pedestrian trajectory prediction enables faster progress in autonomous driving and robot navigation where complex social and environmental interactions involve. Previous models use grid-based pooling or global attention to measure social interactions and use Recurrent Neural Network (RNN) to generate sequences. However, these methods can not extract latent features from temporal and spatial information simultaneously. To address the limitation of previous work, we propose a Self-Centered Star Graph with Attention (SCSG Attention) framework. Firstly, pedestrians' historical trajectories are encoded. Then multi-head attention mechanism plays a role as enhancement of social interaction awareness and simulation of physical attention from human beings. Lastly, the self-centered star graph decoder can aggregate temporal and spatial features and make predictions. Experiments are conducted on public benchmark datasets and measured with uniform standards. Our results show an improvement over the state-of-the-art algorithms by 38% on average displacement error (ADE) and 19% on final displacement error (FDE). Furthermore, it is demonstrated that the star graph has better performance in efficiency of training convergence and ends up with better results in limited time.

Keywords: Pedestrian trajectory prediction · Spatial temporal model · Multi-head attention

1 Introduction

Autonomous driving and robotics that involves human-machine interactions are one of the most promising field of research because there is an unprecedented tendency that let artificial intelligence serve human being and change people

© Springer Nature Switzerland AG 2021
C. S. Jensen et al. (Eds.): DASFAA 2021, LNCS 12681, pp. 422–438, 2021.
https://doi.org/10.1007/978-3-030-73194-6_29

lifestyle. Among all the robotics problems, how to make a machine have a comprehensive understanding of people's motion is a significant issue. Pedestrian trajectory prediction is a branch of the problem from human motion study because it allows robots to plan their own movement to avoid the collision. For example, a self-driving car can make a prediction of pedestrians' movement on the road to make a reasonable adjustment in advance thus avoid the collision. A domestic robot can predict people's trajectory in the room and plan its own movement to minimize the impact on people. Besides, pedestrian trajectory prediction has some important applications in urban city planning and surveillance systems.

Previous work can be classified into four categories: hand-craft rules, grid-based methods, attention-based methods, and graph-based methods. First, the traditional hand-craft rules that simulate human social force [7] have decent results in some circumstances but can not be generalized well on modern datasets because rigid methods are not flexible enough to simulate complex situations in modern datasets. Second, in recent years, Social LSTM [1] is a pioneering work that uses RNN models to make a prediction and also utilizes grid-based pooling to aggregate multiple interactions. After that many different social awareness models are proposed to extract social interactions [3–5]. However, grid-based measurement is not efficient. Sparse grids occupy numerous storage so it needs a large amount of computational power and traversal through grids. Besides, there is a lack of bias on social interaction, which means impacts from different people are considered similarly. Third, to allocate weights to different people and obstacles, global attention models are used in Sophie [14]. However, the drawback of global attention is that it ignores impacts on himself and some potential information from other pedestrians. Forth, recently, SAPTP [6] uses a graph-based method to associate temporal and spatial information and achieve competitive results. Nevertheless, they use a complete graph to extract superfluous features, which will cost redundant computational power.

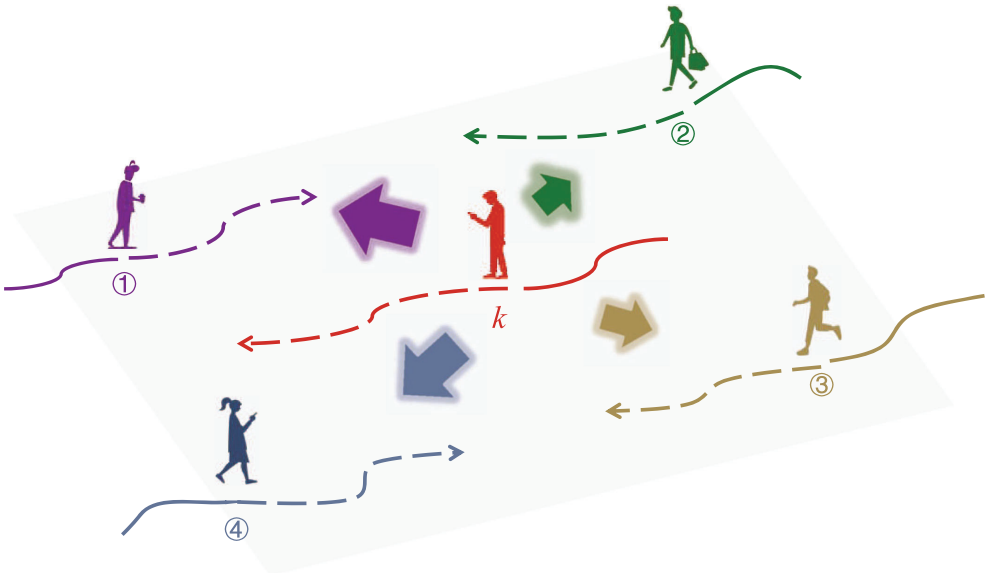


Fig. 1. Attention to different social interaction (Color figure online)

According to the problems mentioned above, challenges mainly come from two factors: (1) how to extract features that represent social interactions is a difficult task. Let's take Fig. 1 as a running example. The pedestrian k in red changes his path mainly because he wants to hide away from pedestrian 1 in purple while he is less influenced by pedestrian 3 in yellow. It shows that he is influenced by others who are not only close to him but also in conflict direction, fast relative speed, etc. All the potential factors can be influential in social interactions. In addition, multiple external impacts including dynamic and static interactions are supposed to be considered at the same time. The variety of social interactions has not been considered by recent studies. (2) How to aggregate temporal and spatial information simultaneously is also a critical problem. Trajectory prediction can be regarded as a two-dimensional sequence generation issue. Therefore, the chronological order of pedestrians' position is essential. Previous works only consider temporal features at decoder, which is not enough to generate future sequences.

To tackle these challenges, a self-centered star graph based on a multi-head attention model is proposed. We use the multi-head attention model to simulate human being's attention in the real world because it indicates people's reflection on the environment and social impacts. For example, when a person walks on an empty street, he probably will not change his direction and walk in a straight line. On the contrary, when a person walks on a crowded street, he will change his direction and speed to avoid other pedestrians. Therefore, the moving behavior of the pedestrian is mainly influenced by how much attention he should pay to the surroundings. That idea inspires us to build an attention-based model to learn environmental influence. The multi-head attention consists of several layers. Every layer can extract useful latent features respectively thus it is a more comprehensive representation of a human being attention. In addition, the novel self-centered star graph is a data structure designed for pedestrian trajectory prediction. It can combine temporal and spatial information because both of them will flow through the graph. We use the graph at the decoder to capture dynamic and static changes in the environment. Besides, it generates attention simulation from a personal perspective and gets rid of redundant calculations.

Contribution of our paper can be summarized as following:

- We propose the utilization of multi-head attention to simulate pedestrians' attention. Multi-head attention can extract different levels of latent features from social interactions. With a more comprehensive feature representation, our model can find the most possible decision made by the pedestrian.
- A self-centered star graph is proposed to capture temporal and spatial features simultaneously. At the same time, it only takes account of the target pedestrian's interaction with nearby people thus it accelerates training speed.
- Our model is conducted on the benchmark datasets and achieves state-of-the-art accuracy and efficiency of convergence. Extensive experiment results show an improvement over that of previous work by 38% on ADE and 19% on FDE.

The remainder of the paper is structured as follows. Section 2 introduces some important notations and a formal definition of the pedestrian trajectory prediction problem. Our SCSG Attention framework as well as its components are illustrated in Sect. 3. Section 4 presents our evaluation of experimental results and a case study. Lastly, the related work and conclusion are shown in Sect. 5 and Sect. 6.

2 Problem Definitions and Important Notations

Definition 1 (Trajectory sample point). A trajectory sample point p is a location in two-dimensional space, and p_i^t represents the sample point at a specific time stamp t of person i.

Each scene of pedestrians is captured at a fixed frequency in videos, where time can be regarded as frame based on videos. Therefore there is a corresponding time sequence $\{t|t = 1, 2, 3, \ldots, n\}$ where n is final frame. In each frame, every pedestrian will be represented by a two-dimension world coordinate $P = \{(x_i^t, y_i^t)|t = 1, 2, 3, \ldots, n \quad i = 1, 2, 3, \ldots, I\}$ where I is the number of distinct pedestrians of all time. And p_i^t denotes pedestrian i's position at time t.

Definition 2 (Trajectory). Trajectory is a sequence of trajectory sample points, ordered by time stamps t. In this problem, pedestrian i's trajectory is a sequence of two-dimensional trajectory sample points: $T_i = [p_i^1, p_i^2, \ldots, p_i^n]$.

Problem Definition. At any frame t, the problem can be defined as following: from observation of target person k's historical trajectory $T_k^h = [p_k^{t-\lambda+1}, \ldots, p_k^t]$ and his neighbor $T_i^h = [p_i^{t-\lambda+1}, \ldots, p_i^t]$ where $i \neq k$, we want to predict the target pedestrian k's future trajectory $\hat{T_k}^f = [p_k^{t+1}, \ldots, p_k^{t+\delta}]$. λ and δ are historical and future length respectively. Specifically, the target person is denoted by index k in this paper. A static object in the street can be treated as a static pedestrian.

This task can also be viewed as a sequence generation problem, where the input sequence corresponds to the observed positions of a person and we want to generate an output sequence. Our goal is to make predictions $\hat{T_k}^f$ as accurate as possible to the ground truth trajectory T_k^f.

3 Methodology

3.1 SCSG Attention Framework

The overview of model architecture can be shown in Fig. 2. There are three components in our framework, namely (1) spatial and temporal encoder, (2) attention mechanism, (3) self-centered star graph decoder. In the beginning, the historical trajectory of target pedestrian k and his neighboring pedestrians i are encoded by temporal and spatial encoder respectively. Hidden states h_k^t and h_i^t

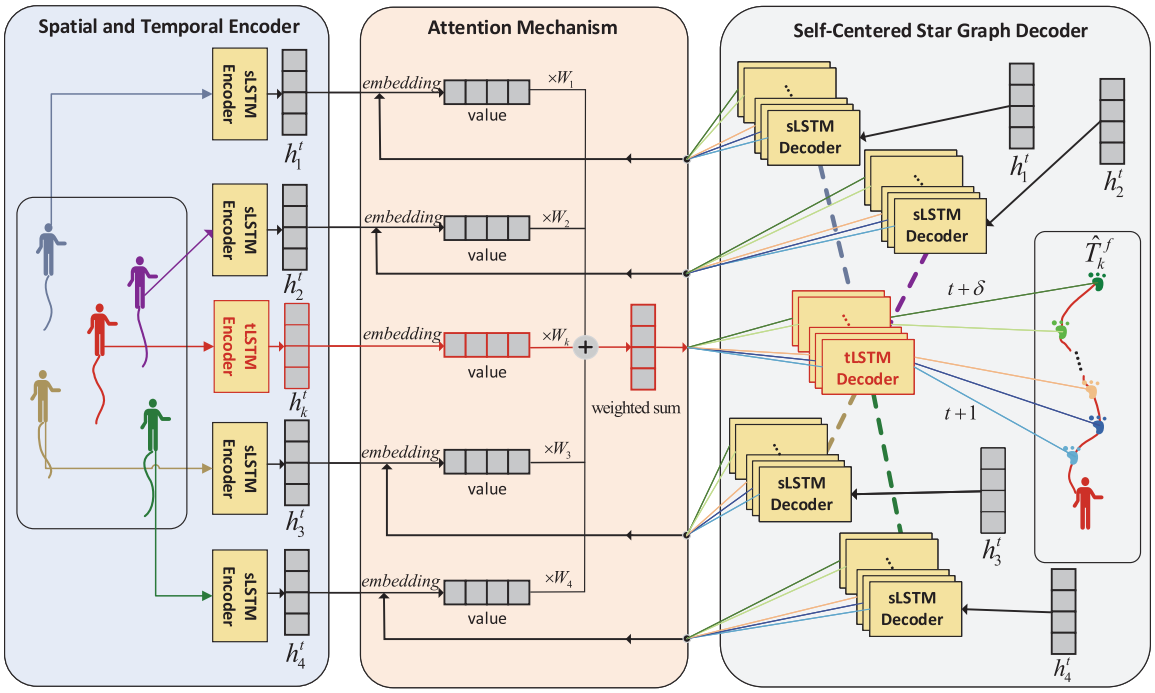


Fig. 2. SCSG attention framework overview

are then entered to our attention mechanism to analog pedestrian k's attention. Finally, the weighted sum of attention vectors is passed through our self-centered star graph decoder to output a predicted location one at a time. At the same time, the neighboring hidden states will be continuously decoded in the graph. Specifically, we calculate attention for every future frame.

3.2 Spatial and Temporal Encoder

Pedestrian location description is based on Cartesian coordinates, thus trajectories in a scene can be shown by Fig. 1. The historical trajectories are represented by a solid line and dash line shows future trajectories. The pedestrian k's historical trajectory contains temporal information and other pedestrian i's historical trajectories are regarded as spatial information. Long Short-Term Memory Networks (LSTM) shows promising functionality in sequence memorization and encoding. For this specific problem, temporal information and spatial information are encoded separately.

For temporal encoding, we defined a dedicated time embedding mapping function to convert historical trajectory T_k^h from locations to a high dimension vector e_k^t as follows:

$$e_k^t = \phi_{temporal}(p_k^t; W) \tag{1}$$

where $\phi_{temporal}(.)$ is a fully connected neural network and p_k^t denotes the location of pedestrian k at frame t, W denotes embedding parameters.

To aggregate historical trajectory features, we define a dedicated temporal LSTM layer to transform temporal embedding e_k^t to a hidden state h_k^t as follows:

$$h_k^t = LSTM(e_k^t, h_k^{t-1}; W) \tag{2}$$

where h_k^{t-1} is the hidden state at last frame, W denotes temporal LSTM parameters. Temporal LSTM layer is executed recursively to obtain the final hidden state h_k^t.

Similarly, a spatial embedding layer is built to transform neighboring pedestrian trajectories T_i^h to high dimension vectors. The embedding layer consists of a fully connected layer. Notably, it does not share parameters with temporal embedding layer but it shares parameters among neighboring pedestrians because neighboring pedestrians together represent context of the target pedestrian. The vector e_i^t is defined as follows:

$$e_i^t = \phi_{spatial}(p_i^t; W) \tag{3}$$

where i is a neighboring pedestrian($i \neq k$) in the scene, p_i^t is location of neighboring pedestrian i at frame t, And W denotes spatial embedding parameters.

We use spatial embedding as input of spatial LSTM in order to incorporate location information of neighboring pedestrians. The spatial LSTM does not share parameters with temporal LSTM. Hidden states h_i^t are defined as follows:

$$h_i^t = LSTM(e_i^t, h_i^{t-1}; W) \tag{4}$$

where i is a neighboring pedestrian($i \neq k$) in the scene, h_i^{t-1} is the pedestrian i's spatial hidden state at previous frame and W denotes spatial LSTM parameters. Spatial LSTM is executed recursively to obtain spatial hidden states of all neighboring pedestrians.

3.3 Attention Mechanism

Human being's attention can allocate bias on different objects thus it makes people focus on useful information. For example, someone who walks on a street will pay more attention to other noticeable pedestrians. Like human beings, attention mechanisms let machines learn useful features thus it makes machines more efficient. In [14], global attention is used to find weights on other pedestrians, but it can not extract features from multiple perspectives. Recently, multi-head attention proposed in [17] vastly boosts the development of the attention mechanism. It is a substitute to simulate physical attention awareness from a personal perspective in our model. To be more specific, multi-head personalized attention is used to mimic physical attention to nearby people in a radius. Therefore, different nearby pedestrians will be measured by unique weights. And multi-head attention can simulate the attention from multiple potential reasoning, which vastly increases the robustness of our model.

The architecture of attention mechanism can be shown in Fig. 3. It shows the calculation process of target pedestrian's attention to himself and his neighboring

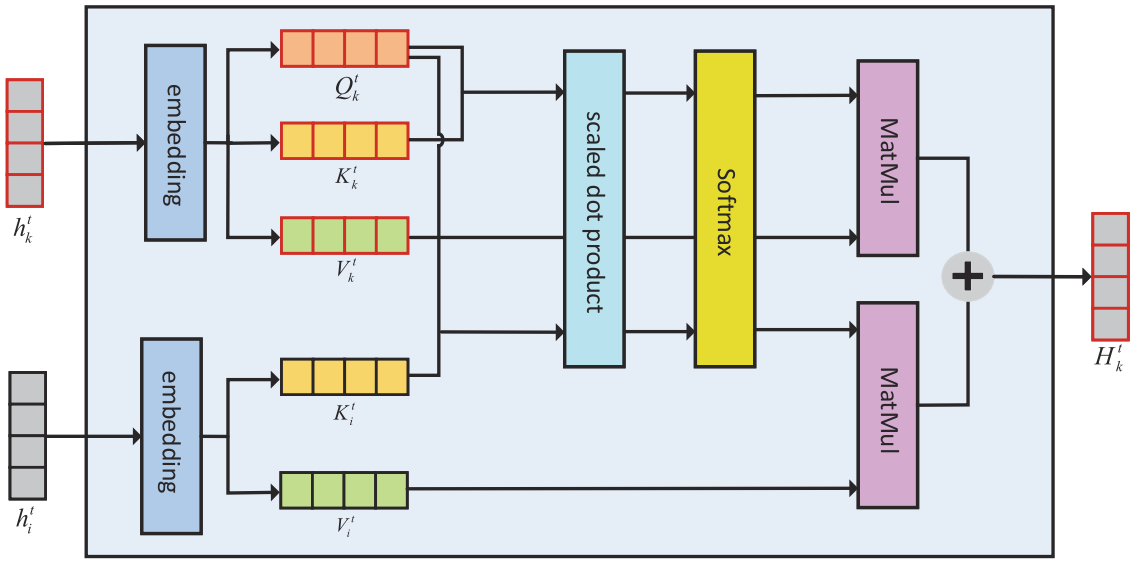


Fig. 3. Attention mechanism

pedestrians in the scene. To calculate attention weights, the output of temporal LSTM h_k^t will be embedded to three vectors namely query vector Q_k^t, key vector K_k^t and value vector V_k^t $(size = d_m)$, which are measured as follows:

$$Q_k^t = \phi_t(h_k^t; W_{qt}) \tag{5}$$

$$K_k^t = \phi_k(h_k^t; W_{kt}) \tag{6}$$

$$V_k^t = \phi_t(h_k^t; W_{vt}) \tag{7}$$

where W_{qt}, W_{kt}, W_{vt} in the embedding function are their parameters respectively.

For outputs from spatial LSTM h_i^t, they will be embedded to key vectors K_i^t and value vectors V_i^t. They are defined as follows:

$$K_i^t = \phi_k(h_i^t; W_{ks}) \tag{8}$$

$$V_i^t = \phi_t(h_i^t; W_{vs}) \tag{9}$$

where W_{ks} and W_{vs} in the functions are embedding function parameters.

For target pedestrian k in the scene, the attention values of latent feature j $Score_j^t$ of pedestrian k can be calculated as follows:

$$Score_j^t = \sum_{i=1}^{n} Softmax(\frac{Q_k^t \times K_i^t}{\sqrt{d_m}}) \times V_i^t \tag{10}$$

where index j indicates index of latent features, user can define how many features to calculate.

Finally, this workflow will be repeated for a user-defined fixed number n. Every output is a layer of attention. The benefit of multi-head attention is that every separated attention will extract a feature from a different perspective.

Different from local attention or global attention, it makes a well-rounded consideration to simulate pedestrians' physical attention. The final attention score F_k^t is defined as follows:

$$F_k^t = W_a \times Concat(Score_1^t, Score_2^t, \ldots, Score_n^t) \tag{11}$$

where W_a is matrix representing a linear transformation.

3.4 Self-centered Star Graph Decoder

LSTM can be a sequence generator since it can decode hidden states and predict results one at a time. But it is hard to capture dynamic changes by naively applying LSTM on some real-world problems that are highly dependent on temporal and spatial features. A spatio-temporal graph-based model was proposed in SAPTP [6] to solve pedestrian prediction. While they use a complete graph in their model, a simplified version namely a self-centered star graph is proposed in our model without sacrificing effectiveness and accuracy. The self-centered star graph decreases the number of edges from $O(n^2)$ to $O(n)$, resulting in a faster convergence speed compared to a complete graph.

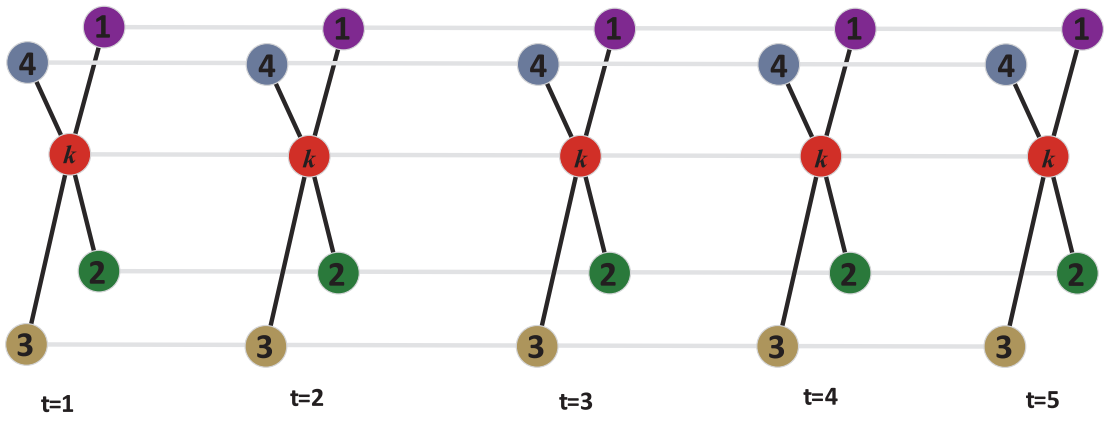


Fig. 4. Self-centered star graph structure

Figure 4 shows the structure of self-centered star graph. It is established in three steps: (1) Add each pedestrian k and i to the vertex set V. So according to the running example in Fig. 1, there are five vertices v_k, v_1, v_2, v_3, v_4 in the beginning. Then add undirected edges from v_k to v_i, denoted by $e(k, i)$. Now a plane self-center star graph is completed (2) Repeat step (1) for $\delta - 1$ times. When $\delta = 5$ (shown in Fig. 4), we get a three-dimensional self-centered star graph. (3) Add undirected grey edges from v_i to v_i and from v_k to v_k respectively from each frame t to $t + 1$, denoted by $e(i, i)$ and $e(k, k)$. That is how the running example in Fig. 1 becomes Fig. 4 topology.

Different edges represent different calculations. From every frame t to $t + 1$, edge $e(i, i)$ represents propagation of spatial information. Specifically, the

neighboring pedestrians hidden states h_i^t will be inputted into a spatial decoder as follows:

$$h_i^t = LSTM(h_i^{t-1}; W) \tag{12}$$

where h_i^{t-1} is the hidden state from previous frame and W denotes the spatial LSTM parameter.

Besides, from every frame t to $t+1$, edge $e(k,k)$ represents propagation of temporal information. Specifically, the target pedestrian hidden states will be decoded by a temporal LSTM as follows:

$$h_k^t = LSTM(F_k^t, h_k^{t-1}; W) \tag{13}$$

where h_k^{t-1} is hidden state of the target pedestrian from previous frame, F_k^t is attention score shown in Eq. 11 and W denotes the temporal LSTM parameters.

At every frame t, edge $e(k,i)$ represents attention from pedestrian k to pedestrian i. The calculation is shown in Eq. 10. After that, the target hidden state will time a matrix W_0 to generate predicted location of the target pedestrian $\hat{p_k^t}$ as follows:

$$\hat{p_k^t} = W_o \times h_k^t \tag{14}$$

Finally, L2 loss is used as our loss function. We have tried both to sample from bivariate Gaussian distribution and to use L2 loss. We find using L2 loss is much beneficial to gradient descent since it boosts the velocity of gradient descent and achieves better results in a shorter time.

4 Experiments and Analysis

In this section, we evaluate our method on several benchmark pedestrian datasets. Besides, our model is compared with selected baselines on two metrics: ADE/FDE.

4.1 Experimental Setup

Datasets: We use two public pedestrian datasets. First, ETH [13] has 750 pedestrians and is divided into two datasets (ETH and hotel) according to two different scenarios. Second, the UCY dataset [10] has 786 pedestrians in total and is divided into three different datasets (zara01, zara02, and univ.) according to different scenarios. Therefore, we used a total of five scenarios to verify our model. These datasets were collected from the real world, including a variety of complex scenes, such as a crowd or two pedestrians walking together. Each pedestrian has a nonlinear trajectory at different speeds.

Evaluation Metrics: According to our baselines, two evaluative metrics are used. The smaller of these two metrics are, the better the model performs.

- Average displacement error(ADE): it calculates the mean square error between all predicted points and ground truth points in a trajectory.

– Final displacement error(FDE): it calculates the distance between the final point of a predicted trajectory and ground truth value of final point.

Evaluation Baselines: We compare our model with the previous competitive models.

– LSTM (vanilla LSTM model): a vanilla LSTM that contain classic encoder-decoder architecture.
– S-LSTM [1]: Social LSTM proposed a grid-based pooling layer, which is designed to model each person via an LSTM with the hidden states being pooled at each time.
– S-GAN [5]: an Generative Adversarial Network(GAN) is used to generate multiple socially-acceptable trajectories. Gupta et al. propose a new grid-based pooling mechanism which encodes the subtle cues for all pedestrians involved in a scene. This model performs well in crowded scenes.
– CF-LSTM [18]: Cascaded Feature-Based Long Short-Term Networks where the feature information of the previous two timestamps is considered as the input of LSTM. Only one pedestrian feature is used in this model.
– SAPTP [6]: a spatio-temporal graph-based model use complete graph to make prediction.
– Global attention model: global attention is used in our model as a comparison to multi-head attention. It only extracts one latent feature and it does not count pedestrian attention to himself.
– Local attention model: local attention is used in our model as a comparison to multi-head attention. Similarly, it only extracts one latent feature. And it generate output attention randomly instead of a weighted sum.

We try our best to reproduce the Cascaded Feature-Based LSTM model (CF-LSTM) following implementation details in the paper [18] and reproduce the vanilla LSTM model to predict the trajectory. Besides, there are three groups of experiments designed to prove our method better than other methods.

Implementation Details: According to all benchmark results, the leave-one-out approach [18] is used to train and validate model parameters. To more specific, every time we train and validate our model on 4 datasets and test on the remaining one. We set $\lambda = 8$ and $\delta = 12$, in other word 3.2 s and 4.8 s respectively. LSTM with 128 units of hidden states is used as encoder and decoder in our model. We use six heads for multi-head attention. Our model is trained with a batch size of 128 for 100 epochs using Adam with a default setting.

4.2 Performance Evaluation

In this subsection, in order to demonstrate the effectiveness of our method, we compare our evaluation metrics with other baselines.

Our Model vs Baselines: Our model is evaluated based on two metrics ADE and FDE against different baselines in Table 1. All baselines use LSTM as the

base model since the classic encoder-decoder architecture is powerful in sequence-related problems. The majority of deep learning models in trajectory prediction is based on vanilla LSTM. As expected, all models perform better than it but CF-LSTM is worse than LSTM in FDE. We try our best to reproduce CF-LSTM and fine-tuned parameters on real-world datasets. It is probably because it only extracts limited features from one person while other models make use of multiple people's trajectory. As shown in Table 1, S-GAN performs better than LSTM, CF-LSTM, S-LSTM, and SAPTP, since it revises the pooling layer in S-LSTM and uses generative modeling to produce multiple possible results. Our model outperforms all other models in ADE and FDE since we use a more well-rounded attention mechanism compared to S-GAN. Furthermore, better aggregation graph is used in our model compared to traditional encoder-decoder methods.

Table 1. Quantitative results of baselines and our models on all datasets

Performance (ADE/FDE)						
Type	Base Model	Pooling	Pooling	Individual	Graph	Graph
Datasets	LSTM	S-LSTM	S-GAN	CF-LSTM	SAPTP	Our Model
ETH	1.41/3.13	1.09/2.35	0.87/1.62	1.36/3.40	1.24/2.35	**0.58/1.47**
HOTEL	0.54/1.38	0.79/1.76	0.67/1.37	0.44/1.22	0.48/0.80	**0.20/0.65**
UNIV	1.47/2.83	0.67/**1.40**	0.76/1.52	1.18/2.86	0.69/1.45	**0.53**/1.53
ZARA1	0.41/1.00	0.47/1.0	0.35/0.68	0.38/1.13	0.51/1.15	**0.20/0.62**
ZARA2	0.34/0.93	0.56/1.17	0.42/0.84	0.33/0.96	0.56/1.13	**0.19/0.60**
Average	0.83/1.85	0.72/1.54	0.61/1.21	0.74/1.91	0.70/1.38	**0.34/0.97**

Effectiveness of Multi-head Attention: Attention mechanism plays an important role in our model, and different attention mechanism has different performance. In this set of experiments, we compare the ADE and FDE of different attention mechanisms. As shown in Fig. 5, ADE and FDE from the multi-head attention model are the lowest among all datasets so the multi-head attention model outperforms other methods evidently. An important reason is that the multi-head attention mechanism can pay attention to the subtle cues of pedestrians around. However, global-attention and local-attention both can only pay attention to partial information, leading to negligence of some important information.

Effectiveness of Different Amount of Heads: The amount of the head represents how many times we count attention. It may affect the effectiveness of our attention mechanism since the number of heads partly determines how many features our model can learn. For example, six multi-head means that there are six attention layers to learn six different latent features while hand-craft rules can only extract one rigid feature. These six results will be concatenated and passed to the next layer. In this set of experiments, we compare the ADE and

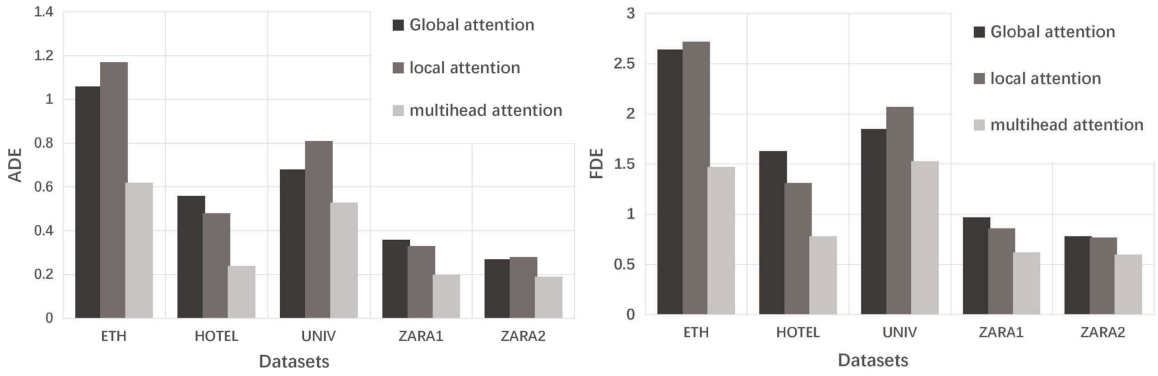


Fig. 5. Comparison of ADE and FDE among attention mechanisms

FDE of different amounts of heads. As shown in Table 2, when the head equals 6, the model outperforms other values of the head. Besides, we only consider the head equal to these values $\{2, 4, 6, 8, 10, 12\}$, the result will be worse with larger number of heads, since it causes overfitting.

Table 2. Multi attention mechanism (Hyparamater)

Datasets	Performance of different head numbers (ADE/FDE)					
	2	4	6	8	10	12
ETH	0.75/2.04	0.64/1.60	**0.58**/**1.47**	0.62/1.51	0.59/1.54	0.61/1.57
HOTEL	0.22/0.87	0.24/0.91	0.20/0.65	0.21/0.72	**0.17**/**0.56**	0.20/0.74
UNIV	0.56/1.59	0.55/1.55	**0.53**/**1.53**	0.54/1.54	0.55/1.53	0.56/1.55
ZARA1	0.21/0.66	0.20/0.64	0.20/**0.62**	0.20/0.63	0.20/0.65	0.20/0.65
ZARA2	**0.19**/0.61	0.20/0.62	**0.19**/**0.60**	0.20/0.64	0.20/0.62	0.20/0.61
Average	0.39/1.15	0.37/1.06	**0.34**/**0.97**	0.35/1.01	**0.34**/0.98	0.35/1.02

Effectiveness of the Self-centered Star Graph: The self-centered star graph exhibits two advantages compared with other methods. First, it is designed to capture spatial and temporal features simultaneously, which is shown in the model architecture part. And the average displacement error comparison to the complete graph proves the effectiveness of the star graph. In the experiment (see Table 3), ADE and FDE are compared to show that the effectiveness of star graph is comparable to that of complete graph. Second, less computation is generated in the star graph, which is expected to produce results in less time. According to our experiments, the complete graph model occupies 95% GPU memory while star graph model occupies 83% GPU memory. A stochastic gradient descent loss graph (see Fig. 6) is presented to prove its velocity. Obviously, a faster decreasing tendency can be shown in the graph. Although the loss of a star graph is higher at the beginning, it finally becomes lower in the limited time. Compared with

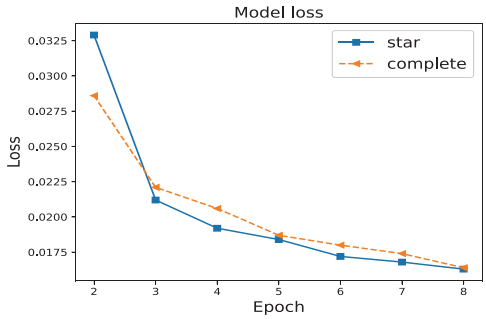


Fig. 6. Loss of star and complete graph

Table 3. ADE and FDE of two graphs

	Performance (ADE/FDE)	
Datasets	Complete graph	Star graph
ETH	0.65/**1.32**	**0.58**/1.47
HOTEL	**0.19**/**0.57**	0.20/0.65
UNIV	**0.51**/**1.47**	0.53/1.53
ZARA1	**0.19**/**0.62**	0.20/**0.62**
ZARA2	**0.13**/0.66	0.19/**0.60**
Average	**0.33**/**0.93**	0.34/0.97

the complete graph, the self-centered star graph can focus on the interaction from valuable people, and also reduce the amount of calculation.

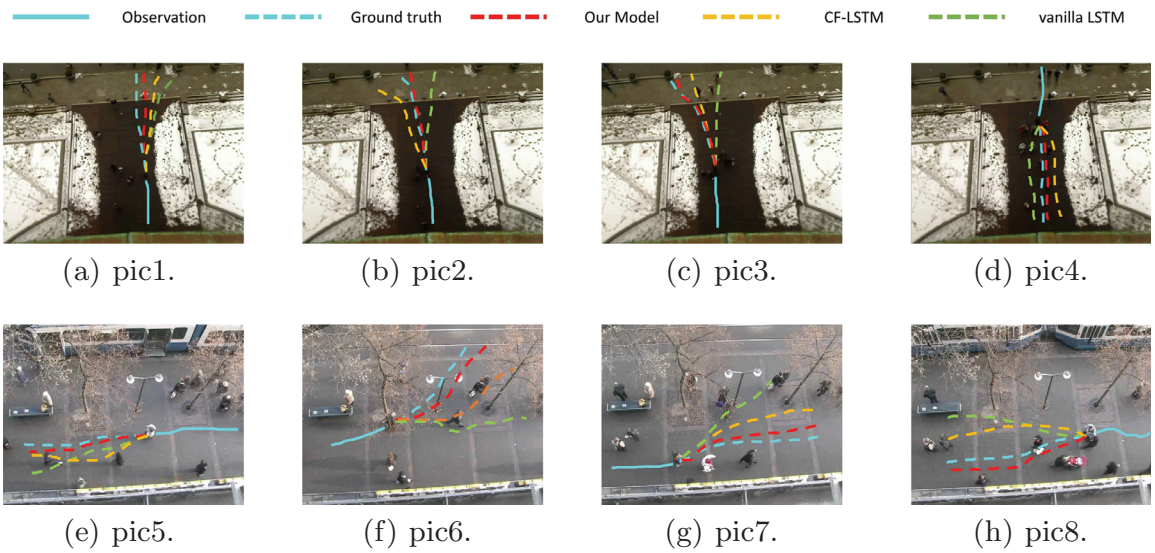
4.3 Case Study

In this section, according to the results of experiments, some of predicted trajectories are visualized in Fig. 7.

In the first row, the background of these pictures is from ETH, and has two characteristics: simple situations where there are interactions only among people, and there are few noticeable obstacles; most pedestrians move in the same direction. In these four pictures, Figure(a) shows the straight trajectory of a pedestrian with fewer interactions from people around, Figure(b) shows the straight trajectory of a pedestrian with more interactions from people around. Figure(c) shows the crooked trajectory of a pedestrian. Figure(d) shows the trajectory of a pedestrian following multiple pedestrians in the same direction. Obviously, our proposed model has a smaller error and the direction of the predicted trajectory is closer to the ground truth.

In the second row, the background of these pictures is from HOTEL, compared to the first row, these pictures involve complex situations, there are many obstacles in the scene, such as a bench, street lamp, trees. Those static objects will be regarded as static pedestrians so they will be taken into consideration. Besides, these pedestrians move in different directions. In these four pictures, Figure(e) shows that a pedestrian moves in a straight trajectory at a normal speed. Figure(f) shows that a pedestrian walks through the road between a tree and a street lamp. Figure(g) shows that a pedestrian wants to pass the pedestrian ahead. Figure(h) shows the predicted trajectory of two parallel pedestrians. It can be seen that our model has better performance than other models in such complicated situations.

Above all, our model can detect environmental and social interactions. It can also incorporate spatial and temporal features simultaneously with help of attention mechanism and the star graph.



(a) pic1. (b) pic2. (c) pic3. (d) pic4.

(e) pic5. (f) pic6. (g) pic7. (h) pic8.

Fig. 7. Visualization of predicted trajectory

5 Related Work

In this section, some important milestones about sequence models, social interaction models, and attention models are introduced. Some of them are used in the real world extensively. State-of-the-art algorithms and techniques are inspired by these previous researches.

5.1 RNN Based Sequence Model

Recurrent Neural Networks(RNNs) are deep learning models used in natural language processing extensively. RNNs are mainly used for sequence processing like machine translation [2,16], image captioning [15] and so on. They are inherently good at sequence memorization and generation since inputs of RNNs are fixed-length sequences and are read step by step. Long Short-Term Memory Networks (LSTM) [8] is a kind of RNNs to avoid gradient exploding and gradient vanishing thus it is capable of encoding more temporal information. With the great success of LSTM in natural language processing [12,19], it is used as temporal information encoded in our model. However, the classic encoder-decoder model is not able to aggregate spatial architecture and temporal information simultaneously. In many real-world applications, problems highly depend on temporal and spatial information. So spatio-temporal LSTM [9,11] are proposed to solve this issue. It uses a spatio-temporal graph to be the abstraction of dynamics information. The edges and vertices are converted to unfolded LSTM layers through shared parameters. The core idea of structured LSTM that can incorporate spatio-temporal encoding is utilized in our model.

5.2 Social Interaction Awareness Model

Pedestrian trajectory prediction has been researched for several decades. Helbing et al. (1995) [7] measured social interaction and inner motivation as social force. The simulations of trajectory prediction are based on some heuristic algorithms. Their algorithms achieved decent results in less complicated circumstances. However, in crowded places, deep learning models tend to perform better [3]. Social LSTM [1] is a pioneering model introducing social interaction in LSTM. Pedestrians' trajectory is described in a grid that can show people relative position and interaction. People in the same grid are then aggregated by pooling layers to obtain synthesized social influence. After Social LSTM, several comparable models are introduced like Convolutional Social Pooling [4], which is a convolutional neural network based on the grid. However, grid-based feature extraction cost a large volume of storage space to cover all interaction around the experiment object. Especially in a sparse environment, which is common in some datasets, the sparse matrix can introduce side effects and redundant storage costs. Since pedestrian trajectory depends on multiple possible factors, Generative Adversarial Networks (GANs) based LSTM was introduced in Social GAN that is able to generate multiple possible results [5]. Their social interaction aggregation is also based on grid pooling. Recently, CF-LSTM [18] predicts pedestrian trajectory without extracting features in social interaction. They make use of the residual network to learn features.

5.3 Attention Model

Attention Mechanism achieves great success in Nature Language Processing (NLP), especially in neural machine translation [16]. It is inherently suitable for sequence generation because it let the generator focus on relevant context instead of considering every information equivalently. In trajectory prediction, pedestrian attention will distribute differently according to different social interactions and spatial situations. In the former models like Social LSTM and Social GAN, weights on every pedestrian are considered equivalently, which is not efficient compared to the attention mechanism. There are three types of attention namely global attention, local attention, and self-attention. Different attention mechanisms will have a substantially different evaluation of social interaction. Haddad et al. [6] use a variant of self-attention to achieve a great result. Their attention is based on historical trajectory, while our model is based on current interaction.

6 Conclusion

In this paper, we focus on predicting the future trajectory of pedestrians in a scene. The self-centered star graph is proposed to make predictions. The pedestrian trajectories will first pass through encoders to become high dimensional vectors. Then these vectors will be extracted latent features by the attention

mechanism. Lastly, a self-centered star graph decoder can decode these features and make predictions. We show the efficiency and effectiveness of our model by experiments. Our model proves to work effectively and try to reconstruct complex situations and social norms in real life as much as possible.

Acknowledgment. This work is supported by NSFC (No. 61802054, 61972069, 61836007, 61832017, 61532018), Alibaba Innovation Research (AIR), scientific research projects of Quzhou Science and Technology Bureau, Zhejiang Province (No.2020D010, No.2020D12) and Sichuan Science and Technology Program under Grant 2020JDTD0007. And We thank Qiyang Lyu for his helpful advise.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social LSTM: human trajectory prediction in crowded spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 961–971 (2016)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
3. Becker, S., Hug, R., Hübner, W., Arens, M.: An evaluation of trajectory prediction approaches and notes on the TrajNet benchmark. arXiv preprint arXiv:1805.07663 (2018)
4. Deo, N., Trivedi, M.M.: Convolutional social pooling for vehicle trajectory prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1468–1476 (2018)
5. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social GAN: socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2255–2264 (2018)
6. Haddad, S., Wu, M., Wei, H., Lam, S.K.: Situation-aware pedestrian trajectory prediction with spatio-temporal attention model. arXiv preprint arXiv:1902.05437 (2019)
7. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Phys. Rev. E **51**(5), 4282 (1995)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
9. Jain, A., Zamir, A.R., Savarese, S., Saxena, A.: Structural-RNN: deep learning on spatio-temporal graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5308–5317 (2016)
10. Leal-Taixé, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B., Savarese, S.: Learning an image-based motion context for multiple people tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3542–3549 (2014)
11. Liu, J., Shahroudy, A., Xu, D., Wang, G.: Spatio-temporal LSTM with trust gates for 3D human action recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 816–833. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_50
12. Liu, P., Qiu, X., Huang, X.: Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101 (2016)

13. Pellegrini, S., Ess, A., Van Gool, L.: Improving data association by joint modeling of pedestrian trajectories and groupings. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 452–465. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15549-9_33
14. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: 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, pp. 1349–1358 (2019)
15. Soh, M.: Learning CNN-LSTM architectures for image caption generation. Department of Computer Science, Stanford University, Stanford, CA, USA, Technical report (2016)
16. Stahlberg, F.: Neural machine translation: a review. J. Artif. Intell. Res. **69**, 343–418 (2020)
17. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
18. Xu, Y., Yang, J., Du, S.: CF-LSTM: cascaded feature-based long short-term networks for predicting pedestrian trajectory. In: AAAI, pp. 12541–12548 (2020)
19. Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing. IEEE Comput. Intell. Mag. **13**(3), 55–75 (2018)