Social graph convolutional LSTM for pedestrian trajectory prediction

Yutao Zhou1 | Huayi Wu1 | Hongquan Cheng1 | Kunlun Qi3 | Kai Hu4 | Chaogui Kang2,5 | Jie Zheng1

1 The State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China
2 School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China
3 School of Geography and Information Engineering, China University of Geosciences (Wuhan), Wuhan, China
4 School of Internet of Things, Jiangnan University, Wuxi, China
5 Center for Urban Science + Progress, New York University, Brooklyn, New York, USA

Abstract
Understanding the movement of pedestrians and predicting their future trajectory can be very important in intelligent transportation systems because accurate pedestrian trajectory prediction will improve the level of autonomous driving technology and reduce traffic accidents. The authors address this problem with a social graph convolutional long short-term memory neural network architecture by considering the movement information of each pedestrian and its interaction with neighbours. Specifically, the authors use a graph to model pedestrian walking state where nodes denote the pedestrian movement information, and edges represent the interactions between pairwise pedestrians. An end-to-end architecture that combines a sequence-to-sequence model with a graph convolutional network to learn the movement features and interaction features is used. To capture the interaction influence on different pedestrians, an emotion gate to refine the learned features and filter out useless information is introduced. A companion loss function to increase the ability of the network to capture ‘walking in groups’ behaviour is further proposed. Through experiments on two public datasets (ETH and UCY), the authors prove that our method outperforms the previous methods.

1 INTRODUCTION

Pedestrian is one of the participants in the traffic system, and the research on pedestrian movement patterns can be applied in many fields, such as autonomous driving, robot navigation, and intelligent tracking systems. As an important part of the intelligent transportation system, autonomous driving technology has made great progress in recent years. Accurate pedestrian trajectory prediction is essential for the autonomous driving system because it can help the system make correct operation strategies, including braking, slowing down, turning, or maintaining the current state. Compared to car movement, pedestrian movement is more free, flexible, and complex, which has a more significant impact on decision-making in autonomous driving, especially in scenarios where people and vehicles are highly mixed, such as residential areas, street markets, and parking lots. In addition, accurate pedestrian trajectory prediction can provide a basis for intelligent traffic management. If the behaviour of pedestrians at intersections can be predicted accurately, potential risks can be identified in advance. Warnings can be sent to pedestrians and nearby vehicles through the transportation infrastructure and vehicle communications system, so the probability of traffic accident happenings will be greatly reduced.

The trajectory of a pedestrian is mainly determined by his/her destination and influenced by the movement of pedestrians around him/her. A critical task in human trajectory prediction is modelling pedestrian interactions. These interactions include walking in groups, turning or decelerating to avoid a collision, and other intricate and subtle movements. Accurately understanding and predicting crowd interactions is challenging because such interactions are related to the social convention, the scenario environment, and even the personality of pedestrians. Limited observed motion trajectories (usually only a few seconds) and simultaneously predicting trajectories of all pedestrians in a scene also make the problem more difficult.
Recently, some data-driven approaches have used the long short-term memory (LSTM) [1] model to capture the movement pattern of pedestrians. To represent the interactions between pedestrians, these approaches apply various pooling mechanisms to share the hidden states in LSTM [2, 3]. However, these approaches only consider nearby pedestrians in a local neighbourhood and give them the same weight, which is not the case. The influence of different pedestrians varies with their distance, relative speed, relative direction, and personal movement style. For example, if a pedestrian is walking forward, someone walking toward him/her usually has a greater impact than someone walking in the same direction as him/her. Other approaches use attention-based methods to capture the relative influences among pedestrians [4, 5]. They assign different weights to different pedestrians according to the movement pattern represented by the hidden states of LSTM. Experiments show that they can obtain better interaction information than the pooling-based model and achieve more accurate predictions.

Although existing approaches have made considerable progress in addressing specific challenges, two factors have been neglected. First, although the interaction between different pedestrians is well considered, the impact of interaction on pedestrians’ movement varies from person to person and needs further processing. Different pedestrians may take different actions for the same situation. Figure 1 gives an example of two different decisions for a possible collision. Figure 1(a) shows one possibility in which pedestrians A and B choose to make a turn and give way to pedestrian C. Another possibility in Figure 1(b) is that they maintain their original moving state and force pedestrian C to make changes. To address this problem, we propose a neural network architecture to refine the interaction features and filter out useless interaction information. Second, walking in groups is a common phenomenon in crowd scenarios, but most studies only consider the interactions among isolated individuals. According to previous research, up to 70% of people in a crowd are actually moving in groups, such as friends, couples, or families walking together. In addition, strangers can also move in groups, especially in extremely crowded scenario [6]. Different from traditional hand-craft model for walking group recognition, we design a special loss function to autonomously identify group walking and encourage our neural network model consider the overall similarity and accuracy of the predicted trajectories of pedestrians in each group.

In this paper, we propose a novel social graph convolutional LSTM model (SGC-LSTM) to address the limitations mentioned above. We use LSTM to encode the observed trajectory of each pedestrian and obtain their hidden states of movement. The graph convolution network (GCN) is used to capture the interactions among all pedestrians in the scene and output the hidden states of interaction. Then, we combine the movement states and interaction states. The combined state is followed by an emotion gate that determines the degree to which the interaction affects pedestrians. Finally, we use another LSTM as the decoder to generate the future trajectory from this state. For more accurate prediction of pedestrians in groups, we introduce the companion loss to add to the original mean square error loss.

The main contributions of this paper are summarised as follows. (i) We propose a novel structure to combine a recurrent neural network (RNN) encoder–decoder with a GCN to capture the movement features and interaction features. (ii) We use an emotion gate to learn the influence of interaction on different pedestrians and filter useless information. (iii) Based on the mean square error of the absolute coordinates, we add the companion loss to increase the ability of the network to capture the ‘walking in groups’ behaviour.

2 | RELATED WORK

Many researchers construct pedestrian behaviour models to better predict their future trajectories. In recent years, deep learning-based methods have also achieved excellent results in pedestrian trajectory prediction research. In this section, we review some related methods that focus on pedestrian trajectory prediction in terms of model categories. The main differences between these methods lie in the mechanism of capturing the interaction information between pedestrians, and the mechanism of fusing the interaction information with motion information. We also introduce the study of pedestrians group walking and point out their importance in pedestrian trajectory prediction.

**Pedestrian interaction capture.** Helbing and Molnar propose the Social Force model [7], which describes the interaction among pedestrians by establishing a motion model of attraction and repulsion. It is applied to automatic robotics navigation [8] and abnormal crowd behaviour detection [9]. Burstedde et al.
use a two-dimensional cellular automaton model to mediate long-range interactions between the pedestrians and simulate pedestrian traffic. Treuille et al. [11] propose a real-time crowd model based on continuum dynamics. Trautman et al. develop Interacting Gaussian Processes (IGP) [12], a non-parametric statistical model based on Gaussian processes to estimate crowd interaction. Antonini et al. [13] propose a Discrete Choice framework for modelling the short-term behaviour of pedestrians. However, most of these methods are based on hand-crafted features or specific rules. For complex scenarios, manual adjustments need to be made to achieve stable results. In recent years, data-driven methods based on RNNs have achieved more accurate results than the above traditional methods.

**Recurrent neural network for trajectory prediction.** RNNs and their variant structures, such as LSTM [1] networks and gated recurrent unit (GRU) [14] are designed for sequence prediction tasks. RNN-based models have achieved good results in machine translation [15], speech recognition[16] and other issues. RNN-based networks can capture the observed sequence patterns and generate sequences according to these patterns. For pedestrian trajectory prediction, RNNs are used to model the motion pattern of pedestrians and predict their future trajectory. Since the independent RNN cannot obtain the interaction information among pedestrians, researchers introduce movement features and previous hidden states to integrate the information of neighbours [2, 3, 17]. Alahi et al. [2] apply a hidden state sharing mechanism among multiple LSTMs through a social pooling layer. Gupta et al. [3] use a novel neural network structure with max-pooling to aggregate information across people. These methods achieve better results than those that only use independent RNN without considering interaction information. However, these methods assume that only the neighbours within a certain distance can affect the pedestrian’s motion and give equal attention to neighbours in different states, which are unreasonable assumptions in most cases.

**Attention-based methods for trajectory prediction.** Attention is a mechanism for improving the performance of the encoder–decoder model based on RNN. Its main idea is to help models assign different weights to each part of the input and extract more critical and important information. Compared with simple RNN, the attention mechanism can make the model address the parts that are helpful for prediction, while ignoring the irrelevant parts so that models make more accurate judgments. Bahdanau et al. [18] have shown that attention-based RNN models are useful for aligning input and output word sequences for neural machine translation. Vaswani et al. [19] propose Self-Attention mechanisms to generate a richer representation of all given inputs. Attention-based models also achieve good results in image and video captioning [20, 21], speech recognition [22] and other fields. In recent years, many attention-based methods have been used to measure the relative importance among pedestrians in trajectory prediction[4, 5, 17, 23]. Vemula et al. [4] propose a social attention model to assign different weights to each pedestrian during prediction. Fernando et al. [5] use a combined soft + hardwired attention model to map the trajectory information from the local neighbourhood to the future positions of the pedestrian of interest.

These works show that attention-based mechanisms can highlight the important pedestrians in a scene. In this paper, we use the self-attention mechanism to capture the influence importance among pedestrians and treat them as edges on a graph with different weights.

**Graph convolution network.** For capturing pedestrians’ interactions, most previous models simply fuse movement hidden states or use max-pooling mechanisms to extract specific features from movement hidden states. These policies fail to obtain associations within the hidden states, resulting in insufficient interaction information. The graph convolution network [24, 25] is a powerful neural network architecture for machine learning on graphs. GCN leverages both node information and graph structure to aggregate feature information from local graph neighbourhoods using neural networks. GCN can capture the dependence of graphs via message passing among the nodes of graphs with an adjacent matrix [26]. The GCN has made significant progress on many tasks, such as traffic speed forecasting [27, 28] and recommender systems [29, 30]. For the trajectory prediction problem, the pedestrians in a scene can be considered nodes on the graph, while the complex interaction among pedestrians can be regarded as edges (see Figure 2). The movement features of pedestrians determine the weights of these edges. Through forward propagation in a convolutional layer, the GCN can share and pass the movement information among pedestrians. In this paper, we use GCN to capture the interaction among pedestrians.

**Group walking behaviour.** Group walking is a common phenomenon of pedestrian behaviour. It refers to the situation that several pedestrians who know each other or strangers with the same destination maintain a constant relative position and walk together for a while. We collectively call such group companion. Weina et al. [31] use hierarchical clustering of relative velocity characteristics to find groups. Lu et al. [32] use an extended floor field cellular automaton model to simulate the behaviour of pedestrian groups and analyse their impact on the whole. These methods have high accuracy in identifying pedestrian groups, and quantifiable indicators can be obtained with the help of fine physical models. However, due to the high complexity of these models, it is difficult to meet the real-time requirements of trajectory prediction, and these models are designed with hand-crafted features according to specific scenarios, which cannot automatically adapt to diverse scenarios. Considering the trade-off between time consumption and
FIGURE 3  An overview of proposed SGC-LSTM architecture. The model consists of three main components: an encoder, a GCN and a decoder. The encoder takes past trajectories of the pedestrians in a scene as input. The GCN takes all the hidden states of the encoder as input and outputs the aggregated state for each pedestrian. The movement feature and interaction feature are concatenated and then updated by the emotion gate. The decoder generates the predicting trajectory based on the updated concatenation.

performance, we use a simple but effective loss function to identify and improve the prediction accuracy of group walking.

3 | METHODOLOGY

Pedestrians’ trajectories are determined by their destination and by their interactions with surrounding pedestrians. The model needs to capture the characteristics of pedestrian movement while considering the interaction information among pedestrians and integrate them into the prediction of future trajectories. In addition, the model should also be able to find companions to better predict the situation of walking in groups. In this section, we first present the structure of our SGC-LSTM model and then introduce two novel components, an emotion gate and companion loss, which help the model generate more accurate trajectories.

3.1 | Problem definition

The problem of pedestrian prediction can be formulated as a multiple sequence generation problem. Given $N$ pedestrians in a scene, our goal is to predict the future positions of these $N$ pedestrians. Formally, our model predicts each pedestrian $i$'s position $\hat{\mathbf{p}}_i = (\hat{x}_i, \hat{y}_i)$ at time step from $T_{\text{obs}} + 1$ to $T_{\text{pred}}$ based on the position of the pedestrian's trajectory $\mathbf{p}_i = (x_i, y_i)$ from time step 1 to $T_{\text{obs}}$.

3.2 | Model architecture

The overview of our SGC-LSTM model is shown in Figure 3. The main structure is a seq2seq model. We use LSTM as the encoder to capture the movement feature of pedestrians and use GCN for modelling social interactions. Another LSTM as the decoder generates the future trajectory using the combination of the movement feature and interaction feature, which are updated by the emotion gate.

Information for movement intention. Each pedestrian has his/her own movement intention, which can be inferred from his/her movement features. Movement features mean the characteristics of pedestrian movement behaviour in a period of time, including the absolute coordinate, speed, and direction. With these features, we can extract pedestrian interactions and further infer the future trajectory.

We use an RNN model to capture movement features from the observed trajectory of each pedestrian. First, we embed the
position coordinate of each pedestrian to a vector using a multi-layer perception (MLP):

\[ \mathbf{f}_i = \phi(x'_i, y'_i; W_{\text{embedding}}), \]

where \( \phi \) is an embedding layer with ReLU non-linearity and \( W_{\text{embedding}} \) is the weight of the layer. Then we use these embeddings as input to the RNN, to encode the movement pattern:

\[ b'_t = \text{RNN}_e(b'_{t-1}, \mathbf{f}_t; W_{\text{encoder}}) \]

where \( b'_t \) is the hidden state of pedestrian \( i \) at time step \( t \) and \( W_{\text{encoder}} \) is the weight of \( \text{RNN}_e \). The weights \( W_{\text{embedding}} \) and \( W_{\text{encoder}} \) are shared among all pedestrians.

The hidden state of \( \text{RNN}_e \) at time step \( T_{\text{obs}} \) captures the moment features of the observed \( T_{\text{obs}} \) time steps, which reflects individual movement intention.

**Information for social interaction.** In a crowded scene, individual movement is often influenced by the neighbouring pedestrian and results in swerve, follow, acceleration or deceleration. We call this influence social interaction, which is an important factor affecting the future trajectory.

We use GCN to capture the social interaction information in our model. Many studies prove that GCN can aggregate node information based on the structure of a graph [33, 34]. As shown in Figure 2, we regard pedestrians in a scene as a graph \( G(V, A) \). The node set \( V \) represents pedestrians in this scene, and \( A \) is the adjacency matrix, which reflects the degree of interaction between every two pedestrians.

GCN takes the node vectors and an adjacency matrix as input. The node vectors are the encoded vectors of movement intention and the adjacency matrix reflects the degree of interaction between pairwise pedestrians. In this problem, it is difficult to define the interaction degree explicitly. We design a self-attention based module (SAM) to capture the influence among pedestrians and generate the adjacency matrix of the GCN. This module is inspired by the self-attention mechanism [19]. First, we feed the hidden state of the encoder of each pedestrian into linear layers and combine the outputs as \( F \):

\[ f_i = \mathbf{\delta}(b'_t; W_\mathbf{m}) \]

\[ F = [f_1, f_2, ..., f_N], \]

where \( \mathbf{\delta} \) is an MLP with weight \( W_\mathbf{m} \) and \( N \) is the number of pedestrians in a scene. \( F \in \mathbb{R}^{N \times d_f} \) and \( d_f \) is the output size of the MLP. Then, the combined features are multiplied by two parameter matrices separately:

\[ K = FW^K, \]

\[ Q = FW^Q, \]

where \( W^K, W^Q \in \mathbb{R}^{d_f \times d_k} \) and \( d_k \) is the attention vector size. Finally, we compute the dot products of \( Q \) with \( K \), divide them by \( \sqrt{d_k} \), and apply a softmax function to obtain the weights:

\[ A = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right). \]

\( A \in \mathbb{R}^{N \times N} \) is the adjacent matrix of graph \( G \). Note that the matrix is a real number matrix and is not a symmetric matrix, which conforms to the fact that the interaction between pedestrians is not symmetric. Considering that pedestrian interaction only occurs within a certain distance, we truncate the adjacency matrix with a threshold. If the distance between two pedestrians is greater than a certain threshold, the corresponding value of their adjacency matrix is set to 0.

After generating the adjacency matrix, we use the graph convolution network with the propagation rule introduced in [25] to aggregate the information of social interaction among pedestrians as defined in

\[ S_{(j+1)} = \sigma \left( D^{-1/2} A D^{-1/2} S_{(j)} W_{(j)} \right). \]

with \( A = \mathbf{I} + \mathbf{I} \), where \( \mathbf{I} \) is the identity matrix and \( \mathbf{D} \) is the diagonal node degree matrix of \( \mathbf{A} \). \( S_{(j)} \) is the aggregated hidden state in layer \( l \) of all pedestrians in a scene. The hidden state with \( l = 0 \) is initialised by the LSTM hidden states at time step \( T_{\text{obs}} \), that is, \( S_0 = [b^0, b^1, ..., b^N] \). \( W_{(j)} \) is the GCN weight in layer \( l \), and \( \sigma \) denotes the activation function ReLU.

**Emotion gate.** Pedestrians’ movement intention and the interaction with pedestrians around them jointly determine their future trajectory; however, due to pedestrians’ different personalities, the two kinds of information have different proportions in making behavioural judgments. To consider this difference, we introduce a feature selection mechanism called emotion gate. The gate is inspired by the structure in LSTM network, which use gate mechanism to determine how much the previous steps hidden state should be kept and how much the current step information should be added to the hidden state. Here we want to select movement information and interaction information through gate mechanism.

First, we concatenate the output of last GCN layer with the hidden state of the encoder LSTM as a merged vector:

\[ m' = [y', \mathbf{f}']. \]

Then, we use the information for the movement intention of each pedestrian to obtain the emotion gate vector, which is calculated as

\[ \mathbf{g}' = \sigma \left( W_m y'_1, y'_2, ..., y'_{T_{\text{obs}}} + b_m \right), \]

where \( W_m, b_m \) are weight and bias. \( \sigma \) denotes the sigmoid function. The gate is a feature selector that learns the characteristics from historical trajectories represented by the encoder hidden
state $h_i$. The selected vector $\hat{h}_i$ is calculated as

$$
\hat{h}_i = g(\bigodot h_i)
$$

where $\bigodot$ denotes an element-wise product operation.

**Prediction for future movement.** Prior research [2] uses the processed hidden state to predict the parameters of a bivariate Gaussian distribution and sample from it to obtain future coordinates. However, it creates difficulties for backpropagation in the training process since the sampling process is non-differentiable [3]. We avoid this by directly predicting the coordinates $[x_{i+1}^\text{pred}, y_{i+1}^\text{pred}]$.

We use $RNN_d$ as a decoder to predict next positions using a concatenated hidden vector. The prediction for future trajectory is given by

$$
b_{i+1}^\text{pred} = RNN_d(\hat{h}_i; W_{\text{decoder}}),
$$

$$
[x_{i+1}^\text{pred}, y_{i+1}^\text{pred}] = \delta(b_{i+1}^\text{pred}; W_f).
$$

where $t = T_{\text{obs}}, T_{\text{obs}+1}, \ldots, T_{\text{pred}}; b_{i+1}^\text{pred}$ is the hidden state of encoder $RNN_d$ at the $T_{\text{obs}}$ time step of pedestrian $i$. $\hat{h}_i$ is the merged vector after element-wise feature selection via the emotion gate. $W_{\text{encoder}}$ is the weight of $RNN_d$ decoder, $\delta$ is an MLP and $W_f$ is its weights.

**Companion loss.** Walking in groups is a common social behaviour of pedestrians. Pedestrians who walk close to each other can be considered companions and often have the same walking direction and destination. The companion can be not only friends walking together, but also strangers with the same motion goal in a certain range. Their trajectories are very similar for a short period of time and the distance between them does not change much, especially in crowded scenes. To encourage the model to address this kind of relationship among pedestrians, we design the companion loss and add it to the mean loss of the model to train the model.

$$
\mathcal{L}_c = \frac{1}{|C|} \sum_{i,j \in C} \sum_{T_{\text{obs}}+1}^{T_{\text{pred}}} \sum \left| D_i(i,j) - \hat{D}_i(i,j) \right|
$$

$$
D_i(i,j) = \sqrt{(x_i^t - x_j^t)^2 + (y_i^t - y_j^t)^2}
$$

$$
\hat{D}_i(i,j) = \sqrt{(\hat{x}_i^t - \hat{x}_j^t)^2 + (\hat{y}_i^t - \hat{y}_j^t)^2}.
$$

$C$ is the set of companion pairs where $i,j \in C$ indicates pedestrians $i$ and $j$ are considered companions and the distance between them is no more than a threshold $S$ from observation time step $t_1$ to $T_{\text{obs}}$, that is, $D_i(i,j) \leq S, t = 1, \ldots, T_{\text{obs}}, |C|$ denotes the number of companion pairs in $C$.

### 3.3 Implementation details

We set the dimension to 32 for the embedding layer to vectorise spatial coordinates before inputting them to the encoder. The encoder and decoder in our implementation are LSTM. We set the dimensions of the hidden state for the encoder to 32 and the decoder to 64. The hidden size of the self-attention module (SAM) to generate the adjacency matrix of GCN is set to 64 and the neighbour distance threshold is set to be 8 m. The number of GCN layers $l$ is 2. Batch normalisation is applied to the input of the GCN embedding layer. We use the Adam optimiser [35] with an initial learning rate of 0.001 and a batch size of 32 to train the model.

### 4 EXPERIMENT

All the experiments are performed under the same hardware environment with one Intel Core i9-9900K CPU @ 3.60 GHz and one NVIDIA 2080Ti GPU, and under the same software environment with the Ubuntu 18.04 operating system and the open source machine learning framework PyTorch v1.3.0.

#### 4.1 Datasets and metrics

**Datasets.** We evaluate our method on two publicly available datasets: ETH [36] and UCY [37]. These datasets consist of real-world human trajectories with rich human–human interaction scenarios such as walking in groups, groups crossing each other, joint collision avoidance and non-linear trajectories [36]. The ETH dataset consists of two scenarios named ETH and HOTEL. The UCY dataset includes two scenarios named ZARA-01, ZARA-02 and UCY. These five sets of data have four different scenes that consist of 1536 pedestrians. We convert all the data to real-world coordinates and interpolate them to obtain values every 0.4 seconds.

**Evaluation metrics.** Similar to prior work [2, 3], we evaluate the prediction error with two different metrics:

1. **Average Displacement Error (ADE):** The mean square error between the ground truth and our prediction over all predicted time steps.

2. **Final Displacement Error (FDE):** The distance between the predicted final destination and the true final destination at the end of the prediction period $T_{\text{pred}}$.

**Evaluation methodology.** We follow the leave-one-out evaluation methodology in [2], train on four sets and test on the remaining set. We observe the trajectory for eight time steps (3.2 seconds) and predict the trajectory of the following 12 time steps (4.8 seconds).

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

1. Linear: A linear regressor that estimates linear parameters by minimising the least square error.

2. CA-based: This method is proposed by Burstedde et al. [10]. This model uses a two-dimensional cellular automaton model to simulate pedestrian traffic.

3. LSTM: A simple LSTM with no extra structure.
4. S-LSTM: This method is proposed by Alahi et al. [2]. Each person is modelled via an LSTM with the hidden states being pooled at each time step using the social pooling layer.

5. SGAN: This method is proposed by Gupta et al. [3]. This method uses generative adversarial networks to train an encoder–decoder and uses a different pooling layer from S-LSTM.

6. Sophie: This method is proposed by Sadeghian et al. [17]. Sophie consists of three key modules including: A feature extractor module, an attention module, and an LSTM-based GAN module.

Note that the best score in the SGAN and Sophie is obtained by sampling 20 times and choosing the results that best match the ground truth, while the other methods only produce one prediction. The method Sophie also use the environment information.

4.2 Quantitative evaluation

Comparison with baselines. Table 1 shows the results of our method and baselines. All of these methods are evaluated with the same dataset and metrics. Our method outperforms all baseline methods in datasets ETH-hotel, UCY-zara02, and UCY-univ in terms of ADE and FDE. The CA-based method does not perform well on several datasets. The reason is that the cell size limits the prediction accuracy and the simulation of pedestrian interaction is too simple to deal with the complex situations in these scenarios. The LSTM model performs much worse than our model because it only captures the individual movement and does not obtain the interaction information among pedestrians. But it performs better than the linear model because LSTM can produce more complex trajectories.

Our model significantly outperforms S-LSTM and SGAN. These two models both use pooling mechanism to capture the interaction and only consider the pedestrians in the neighbouring area, whereas our model considers a larger range and gives different attention based on their degree of interaction. The SGAN model aims to generate socially acceptable trajectories. To achieve better results, the SGAN samples 20 times (called SGAN-20V-20 in original paper) and chooses the best one as its final result. However, SGAN dose not perform well with only a single sample [3]. Compared to SGAN, the average error metrics of our method in ADE and FDE are reduced by 19.0% and 14.4%, respectively, producing only one prediction. For the attention-based method Sophie, in addition to the trajectories, it also needs the scene image as input. Our model increases the performance relative to Sophie for 13.0/12.2(6%), with only a single prediction and without environment information.

Component analysis for the emotion gate. To verify the improvement by the emotion gate, we remove this component from our model and leave the other settings unchanged (SGC-LSTM_1 in Table 1). As shown in Table 1, the original SGC-LSTM model outperforms the model without the emotion gate(SGC-LSTM_1). Specifically, the metric ADE of origin SGC-LSTM decreased by 6.0%, and the metric FDE decreased by 5.6%. The results verify the effectiveness of the emotion gate.

Component analysis for companion loss. SGC-LSTM_2 is another variant of our model. The loss function of SGC-LSTM_2 is the basic mean square error loss of absolute coordinates without the companion loss addition. The other components and parameters of the model remain unchanged. The experimental results are represented in Table 1. Utilisation of the companion loss achieves an improvement of 7.8% in ADE and 8.1% in FDE. The increases in dataset ETH-univ and ETH-hotel are significant because there are more cases of walking in groups, and most of their trajectories are consistent in these datasets. In these cases, maintaining the similarity of the companion’s trajectories during the prediction process can contribute to improving the model performance.

Time consumption. Considering the application of trajectory prediction, real-time is an important indicator of the model. We compared the prediction time of our model with three baselines: LSTM, S-LSTM and SGAN. As shown in Table 2, the
result is consistent with the complexity of these models. LSTM performs the fastest for its simplicity. Our method is slower than S-LSTM and SGAN (see Table 2) due to the graph structure for calculating the interactions of pedestrians. Although our model is slightly slower than other baselines, the predicting speed of 930.9 trajectories per second of our model reached the requirement of real-time applications.

4.3 Qualitative evaluation

Social interaction. To analyse the learned interaction information, we consider three real scenarios where people have complex interactions, including meeting other pedestrians, following other pedestrians, and bypassing standing or slow pedestrians. We compare our model with LSTM and S-LSTM, and the qualitative results are shown in Figure 4. The LSTM model does not have interaction information and is unable to respond to a possible collision. S-LSTM generates a more reasonable result but still deviates from ground truth. Our model performs better than others on average and predicts more accurate and socially acceptable trajectories in most scenarios. However, for the sudden turn shown in Figure 4(b), all three models failed in prediction because of its unpredictability. These results prove that considering global interaction and assigning different weights to pedestrians in different states will make the predicted trajectory more accurate.

Attention weights. To analyse the effect of the interaction component of our model, we visualise the GCN adjacency matrix value, which is generated by the SAM, to observe the relative weight of each pedestrian to a specific pedestrian in a scene. We consider several crowd scenarios among the datasets, and the results are shown in Figure 5. Note that the pedestrians whose trajectories end with ★ are the target pedestrians of concern. In Figure 5(a), the moving pedestrian gains more attention than the stationary pedestrian. In Figure 5(b), the model assigns more attention to pedestrians who walk in the opposite direction than pedestrians who walk in the same direction.

Furthermore, as shown in Figure 5(c), the model gives local pedestrians higher weight compared to others far away. Besides, the model gives the nearest pedestrian (red) less importance than the pedestrians (green and orange) far away, because the pedestrian (red) who is nearest to the target person is the target person’s companion. They are all walking towards the same destination and it is unlikely that they will influence each other. But the pedestrians in front of them (green and orange) may block their way. So the two pedestrians will have a more impact on the target people than his/her companion. Therefore, the weight given by the model is reasonable. In Figure 5(d), pedestrians are almost equally important as they are at almost the same velocity, and they are all far away from the target pedestrian on the left. Consistent with this situation, the model gives them similar weights. These successful cases show the ability of the SAM in our model to assign reasonable weights to neighbouring pedestrians according to their movement states.

Companion behaviour capture. We also draw several samples to verify our model’s ability to capture companion

---

**TABLE 2** The speed comparison of with several baselines models

|                | LSTM  | S-LSTM | SGAN  | SGC-LSTM |
|----------------|-------|--------|-------|----------|
| Time Usage (s) | 14.08 | 17.13  | 20.62 | 30.09    |
| Rate           | 1x    | 1.21x  | 1.46x | 2.14x    |
| Speed (trajectories/s) | 1989.3 | 1635.1 | 1358.4 | 930.9    |

Note. These results are obtained on the same datasets and the same hardware environment.
behaviour with the help of the companion loss component. Figure 6(b) and (d) shows examples of our model successfully capturing the movement of walking in groups. Figure 6(a) and (c) is the corresponding results generated by the model without companion loss. We can see that the original model with companion loss detects the possible group and maintains a more similar distance among them, which results in a more accurate prediction than the model without companion loss. Our model identifies companions based on whether the distance between pedestrians is less than the threshold. However, for a more accurate companion identification, more behavioural information, such as consistency of speed, should be considered, which is our future work.

5 | CONCLUSION AND FUTURE WORK

In this paper, we present a novel model SGC-LSTM which is a combination of the LSTM encoder–decoder and GCN. We use the LSTM encoder to capture the observed movement information of pedestrians and the GCN to capture the interaction information among pedestrians. The adjacent matrix is generated by a self-attention based module. We also propose a mechanism called the emotion gate to measure the degree to which pedestrians are affected by others. The prediction is generated by the decoder LSTM using the combination of movement information and interaction information refined by the emotion gate. In addition, to learn the characteristics of group movement.

FIGURE 5 Pedestrians’ relative importance to the target pedestrians in 4 crowd scenarios: (a), (b), (c) and (d). The trajectories end with ★ are the trajectories of target pedestrians. The solid dots on the trajectory represent locations of the pedestrians at previous time steps. The circles around the current position of each pedestrian represent importance with the radius proportional to the corresponding adjacency matrix value.

FIGURE 6 Two examples that successfully capture the movement of walking in groups. The trajectories generated by the model with companion loss ((b) and (d)) are more accurate than those generated by the model without companion loss ((a) and (c)).
and improve prediction accuracy, we introduce companion loss which encourages the network to produce more similar predictions for pedestrians walking in groups. Experiments show that our proposed method significantly improves prediction accuracy on two publicly available datasets.

In future explorations, more complex graph convolution network can be used to capture the interaction between pedestrians. For companion recognition, we expect to design more comprehensive discrimination rules to improve the prediction accuracy of companion pedestrians. For further elicitation and analysis of pedestrian movement intention, the anticipatory networks [38] can be applied to define behaviour strategies and coordinate the movement intention of multiple pedestrians.

REFERENCES

1. Alahi, A., et al.: Social LSTM: human trajectory prediction in crowded spaces. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 961–971.IEEE, Piscataway, NJ (2016)
2. Antonini, G., et al.: Discrete choice models of pedestrian walking behavior. Transport. Res. B 40(8), 667–687 (2006)
3. Babdanau, D., et al.: Neural machine translation by jointly learning to align and translate. 3rd International Conference on Learning Representations. (2015) San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings
4. Bruna, J., et al.: Spectral networks and locally connected networks on graphs. 2nd International Conference on Learning Representations. (2014) Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings
5. Burstedde, C., et al.: Simulation of pedestrian dynamics using a two-dimensional cellular automaton. Physica A 295(3-4), 507–525 (2001)
6. Chorowski, J.K., et al.: Attention-based models for speech recognition. Adv. Neural. Inf. Process. Syst. 577–585 (2015)
7. Chung, J., et al.: Empirical evaluation of gated recurrent neural networks on sequence modeling. Preprint, 2014, arXiv:1412.3555 [cs.NE]
8. Chung, J., et al.: A recurrent latent variable model for sequential data. Adv. Neural. Inf. Process. Syst. 2980–2988 (2015)
9. Fernando, T., et al.: Soft+ hardwire attention: an LSTM framework for human trajectory prediction and abnormal event detection. Neural Netw. 108, 466–478 (2018)
10. Ge, W., et al.: Vision-based analysis of small groups in pedestrian crowds. IEEE Trans. Pattern Anal. Mach. Intell. 34(5), 1003–1016 (2012)
11. Graves, A., et al.: Speech recognition with deep recurrent neural networks. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6645–6649.IEEE, Piscataway, NJ (2013)
12. Gupta, A., et al.: Social GAN: socially acceptable trajectories with generative adversarial networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2255–2264.IEEE, Piscataway, NJ (2018)
13. Helling, D., Molnar, P.: Social force model for pedestrian dynamics. Phys. Rev. E 51(5), 4282 (1995)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. 9(8), 1735–1780 (1997)
15. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. 3rd International Conference on Learning Representations. (2015) San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings
16. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. International Conference on Learning Representations. (2017) Toulon, France, April 24–26, 2017, Conference Track Proceedings
17. Larson, A., et al.: Crowds by example. Comput. Graphics Forum. 26, 655–664 (2007)
18. Li, Y., et al.: Diffusion convolutional recurrent neural network: data-driven traffic forecasting. International Conference on Learning Representations. (2018) Vancouver, BC, Canada, April 30 – May 3, 2018, Conference Track Proceedings
19. Li, Y., et al.: Gated graph sequence neural networks. 4th International Conference on Learning Representations. (2016) San Juan, Puerto Rico, May 2–4, 2016, Conference Track Proceedings
20. Lu, L., et al.: A study of pedestrian group behaviors in crowd evacuation based on an extended field floor cellular automation model. Trans. Res. C 81, 317–329 (2017)
21. Mehran, R., et al.: Abnormal crowd behavior detection using social force model. 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 935–942.IEEE, Piscataway, NJ (2009)
22. Monti, F., et al.: Geometric matrix completion with recurrent multi-graph neural networks. Adv. Neural. Inf. Process. Syst. 3697–3707 (2017)
23. Moussaid, M., et al.: The walking behaviour of pedestrian social groups and its impact on crowd dynamics. PloS One 5(4), e10047 (2010)
24. Pellegrini, S., et al.: You’ll never walk alone: modeling social behavior for multi-target tracking. 2009 IEEE 12th International Conference on Computer Vision, 261–268.IEEE, Piscataway, NJ (2009)
25. Sadeghian, A., et al.: Sophie: an attentive GAN for predicting paths compliant to social and physical constraints. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1349–1358.IEEE, Piscataway, NJ (2019)
26. Scarselli, F., et al.: The graph neural network model. IEEE Trans. Neural Netw. 20(1), 61–80 (2009)
27. Skulimowski, A.M.J.: Anticipatory networks. In: Poli, R. (ed.) Handbook of Anticipation, pp. 995–1030.Springer International Publishing, Cham (2019)
28. Su, H., et al.: Forecast the plausible paths in crowd scenes. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, pp. 2772–2778.(2017) Melbourne, Australia, August 19–25, 2017
29. Trautman, P., et al.: Robot navigation in dense human crowds: statistical models and experimental studies of human–robot cooperation. Int. J. Rob. Res. 34(3), 335–356 (2015)
30. Trautman, P., Krause, A.: Unfreezing the robot: navigation in dense, interacting crowds. 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 797–803.IEEE, Piscataway, NJ (2010)
31. Treuille, A., et al.: Continuum crowds. ACM Trans. Graph. 25, 1160–1168 (2006)
32. Vaswani, A., et al.: Attention is all you need. Adv. Neural. Inf. Process. Syst. 5998–6008 (2017)
33. Vemula, A., et al.: Social attention: modeling attention in human crowds. 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 1–7.IEEE, Piscataway, NJ (2018)
34. Xu, K., et al.: Show, attend and tell: neural image caption generation with visual attention. Proceedings of the 32nd International Conference on Machine Learning, pp. 2048–2057.ACM, New York, NY (2015)
35. Yao, L., et al.: Describing videos by exploiting temporal structure. Proceedings of the IEEE International Conference on Computer Vision, pp. 4507–4515.IEEE, Piscataway, NJ (2015)
36. Ying, R., et al.: Graph convolutional neural networks for web-scale recommender systems. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 974–983.ACM, New York, NY (2018)
37. Yu, B., et al.: Spatio-temporal graph convolutional networks: a deep learning model for traffic forecasting. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, pp. 3634–3640 (2018) July 13–19, 2018, Stockholm, Sweden
38. Zhou, J., et al.: Graph neural networks: review of methods and applications. Preprint, 2018, arXiv:1812.08434 [cs.LG]