STGAT: Modeling Spatial-Temporal Interactions for Human Trajectory Prediction

Yingfan Huang[1,2], HuiKun Bi[1], Zhaoxin Li[1], Tianlu Mao[1]*, Zhaoqi Wang[1]
[1]Beijing Key Laboratory of Mobile Computing and Pervasive Device,
Institute of Computing Technology, CAS, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
{huangyingfan, bihuikun, lizhaoxin, ltm, zqwang}@ict.ac.cn

Abstract

Human trajectory prediction is challenging and critical in various applications (e.g., autonomous vehicles and social robots). Because of the continuity and foresight of the pedestrian movements, the moving pedestrians in crowded spaces will consider both spatial and temporal interactions to avoid future collisions. However, most of the existing methods ignore the temporal correlations of interactions with other pedestrians involved in a scene. In this work, we propose a Spatial-Temporal Graph Attention network (STGAT), based on a sequence-to-sequence architecture to predict future trajectories of pedestrians. Besides the spatial interactions captured by the graph attention mechanism at each time-step, we adopt an extra LSTM to encode the temporal correlations of interactions. Through comparisons with state-of-the-art methods, our model achieves superior performance on two publicly available crowd datasets (ETH and UCY) and produces more "socially" plausible trajectories for pedestrians.

1. Introduction

As a challenging task, the human trajectory prediction has attracted considerable attention in computer vision [42, 23, 28, 21, 24, 39, 20] and robotics [5, 19] fields over the past few years. Modeling the complex and diverse interactions among humans is critical and challenging in trajectories prediction, while the hand-crafted energy functions adopted in earlier works [14, 21] failed to build crowd interactions among pedestrians in crowded spaces.

Recently, some LSTM-based (Long-Short Term Memory) methods were proposed to capture the dynamic interactions of pedestrians, where the latent motions represented with the hidden states of LSTMs are shared by various mechanisms including "pooling" [1, 12] and "attention" [33, 10, 13], etc. The "pooling" scheme exploits social pooling on occupancy maps to collect the latent motion dynamics of pedestrians involved in a local neighborhood or the whole scene. Different from the restriction of the local neighborhood assumption, the "attention" mechanism is helpful to encode the relative influences and the potential spatial interactions among pedestrians, due to the unequal importance of the neighboring pedestrians contributing to the trajectories prediction. Compared with the "pooling" scheme, by assigning the different and adaptive importance to the pedestrians, attention-based models can get a better understanding of the crowd behaviors based on spatial interactions.

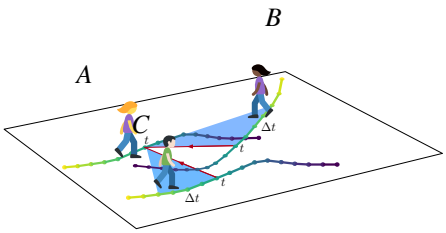

Figure 1. Illustration of the interactions in the crowd. The same color on trajectories means the same time-step. Two red arrows indicate the spatial interactions from pedestrian B and C to A at time t respectively. The blue sector domains are the continuous effects from $t-\Delta t$ to t.

However, although the diverse aspects have been well investigated, a factor was neglected in previous works. Besides the spatial interactions at the same time-step, the temporal continuity of interactions in the crowd is necessary. As shown in Fig. 1, the effects of the spatial interactions from pedestrian B and C at time t have been well-considered in the existing trajectories prediction works. However, due to the continuity and forward-looking nature of human motion, pedestrians need to consider others' historical movements to determine their current motion behavior for avoid-

*Corresponding author.

ing potential collisions in the future. For example, when pedestrian A plans the trajectory, the interactions from $t-\Delta t$ to t from the pedestrian B and C should be taken into consideration. Thus, the temporal correlations of interactions in the crowd play an important role.

To address the limitations mentioned above, we build a novel Spatial-Temporal Graph Attention network (called STGAT), where the spatial and temporal interactions among pedestrians are encoded, respectively. The spatial interactions at one time-step are captured by the Graph Attention (GAT) scheme [32], which models over all the pedestrians involved in the scene. After assigning the different importance on pedestrians, an extra LSTM is used to capture the temporal correlations of interactions. By aggregating all the spatial and temporal interactions among all the pedestrians, the future trajectories are generated by our sequence-to-sequence (seq2seq) architecture. To model the multimodal movement, we adopt the variety loss [12] to produce multiple socially plausible outputs.

Contributions: We present a novel framework (called STGAT) to forecast the trajectories of humans. First, we explicitly model the temporal correlations of interactions by adopting an extra LSTM. To the best of our knowledge, the continuity of interactions has never been considered separately. Second, we model the spatial interactions among pedestrians by using GAT to aggregate hidden states of LSTMs. This paper is the first attempt to combine GAT (graph attention network) with LSTM in the context of modeling pedestrian motions. Experimental results demonstrate that graph attention network can assign reasonable importance to neighbors, and our model can predict reasonable trajectories in different scenes.

2. Related Work

Crowd Interaction The pioneering model for pedestrian dynamics was proposed by Helbing *et al*. [14]. Their Social Force model uses attractive forces to guide pedestrians toward their destinations, and repulsive forces to avoid collisions. Over the past decades, this model has been extended and modified by several approaches [20, 39, 21]. Most of these Social Force based models attempt to learn the parameters of the force functions from real-world crowd datasets. But experiments in [1] have shown that only attractive and repulsive forces can not model complex crowd interactions. There are also other hand-crafted features based models, where Antonini *et al*. used a Discrete Choice framework [3], Treuille *et al*. proposed continuum dynamics [31], and further, there are a series of topics models [35, 15, 9]. However, all these models mentioned above rely on hand-crafted energy potential functions, which limit the performance of prediction accuracy. Recently, there are many deep learning based models, Yi *et al*. [41] proposed Behaviour-CNN, which uses CNN to model crowd interactions. Alahi *et al*. [2] encoded the human-human interactions into a "social" descriptor. Vemula *et al*. [33] proposed a novel spatial-temporal graph, which uses an attention module to merge information from different edgeRNNs. Xu *et al*. [38] used a softmax way to assign different weights to other pedestrians based on spatial affinity. In the past two years, the RNN-based models have achieved great success [1, 33, 12, 13, 16, 38], all these methods use different ways to share the hidden states of RNNs to model interactions between pedestrians in crowded scenes.

Recurrent Neural Networks for Sequence Prediction Sequence prediction problem involves using historical sequence information to predict future values in the sequence. Recurrent Neural Networks, like Long Short-Term Memory (LSTM) networks, are designed for sequence prediction problem. They have achieved great success in many sequence prediction tasks, e.g., speech recognition [7, 11], machine translation [4, 6, 30] and image captioning [8, 26, 37]. The approaches by [1, 29, 38] have proved the success of LSTM for modeling the motion pattern of each pedestrian. However, vanilla LSTM ignores the crowd interactions. To tackle this problem, several attempts have been made to jointly reason across multiple people. Alahi *et al*. [1] used a "social" pooling layer which allows the LSTMs of spatially proximal sequences to share their hidden states. Gupta *et al*. [12] used a "pooling module" in the Generator to aggregate information across people. Xu *et al*. [38] used LSTMs as "motion encoder module" to handle only temporal information, and another module called "location encoder module" is adopted to model spatial interactions.

Sequence to Sequence Model Seq2seq model was introduced by Sutskever*et al*. [30]. It aims to map a fixed length input with a fixed length output, where the two lengths may be different. The seq2seq model and its variants are considered as the best solution for many complex tasks, e.g., machine translation [36], speech recognition [22] and video captioning [34]. Our problem is to predict the future trajectories of all pedestrians given the observed trajectories, while the seq2seq model is designed for generating new sequences based on existing sequences, which is just right for our problem. Thus, we adopt seq2seq as our main architecture.

Graph Neural Network Graph neural networks (GNNs) are powerful neural network architecture for machine learning on graphs. Recent years, systems based on graph convolutional network (GCN) [25] and gated graph convolution neural network (GGNN) [18] have demonstrated ground-breaking performance on many tasks like modeling physics system, learning molecular fingerprints, predicting protein interface [43]. Recently, some methods [27] [40] in the field of action recognition have made significant progress by applying GNNs to spatial-temporal

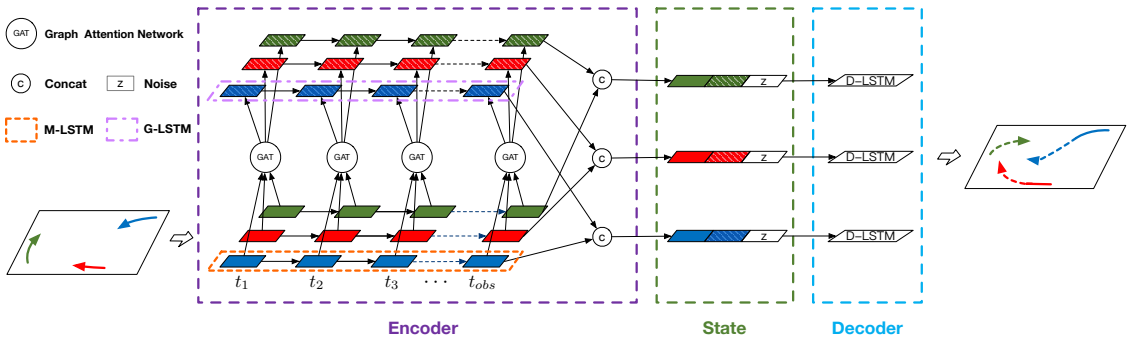

Figure 2. The architecture of our proposed STGAT model. The framework is based on seq2seq model and consists of 3 parts: Encoder, Intermediate State and Decoder. The Encoder module includes three components: 2 types of LSTMs and GAT. The Intermediate State encapsulates the spatial and temporal information of all observed trajectories. The Decoder module generates the future trajectories based on Intermediate State.

data. Among these approaches, Veličković *et al*. [32] proposed Graph Attention Network (GAT). It allows for (implicitly) assigning different importance to different nodes within a neighborhood without costly matrix operations. GAT has achieved or matched state-of-the-art results across several benchmarks for graph-related tasks. For our problem, the complex interactions can be modeled using GAT, where pedestrians in the crowded scene can be considered as nodes on the graph at every time-step, and the existence of interactions between pedestrians can be described as graph edges.

3. Method

Our goal is to forecast the trajectories of the pedestrians involved in a scene. In this section, we present our seq2seq-based STGAT model (as shown in Fig. 2). There are three components in the encoder: an LSTM-based pedestrian trajectory encoding module, a GAT-based module for modeling the spatial interactions, and an LSTM-based module for capturing the temporal correlations of interactions.

3.1. Problem Definition

We assume there are N pedestrians involved in a scene, represented as $p_1, p_2, \ldots, p_N$. The position of pedestrian p_i $(i \in [1,N])$ at time-step t is denoted as $S_i^t = (x_i^t, y_i^t)$. Then given S_i^t of pedestrian $i = 1,2,\ldots,N$ at time-steps $t = 1,\ldots,T_{obs}$, our goal is to predict the future positions S_i^t at time-step $t = T_{obs}+1,\ldots,T_{pred}$.

3.2. Trajectory Encoding for One Pedestrian

Each pedestrian has his(or her) motion pattern, including different gait, preferred speed, and acceleration. LSTM has been proven to successfully capture the historical motion state of a single pedestrian [1, 29, 38]. Similarly, we use one LSTM for each pedestrian to get the motion state. We denote this LSTM by M-LSTM (LSTM for motion encoding).

In our implementation, we first calculate the relative position of each pedestrian to the previous time-step:

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

Then the relative position is embedded into a fixed-length vector e_i^t for every time-step, and these vectors are used as inputs to the LSTM cell as follows:

$$e_i^t = \phi(\Delta x_i^t, \Delta y_i^t; W_{ee}) \tag{2}$$

$$m_i^t = \text{M-LSTM}(m_i^{t-1}, e_i^t; W_m) \tag{3}$$

where the function $\phi(\cdot)$ is an embedding function. W_{ee} is the embedding weight. m_i^t is the hidden state of the M-LSTM at time-step t. W_m is the weight of the M-LSTM cell. These parameters are shared among all the pedestrians in the scene.

3.3. GAT-based Crowd Interaction Modeling

Naive use of one LSTM per person does not capture human-human interactions. In order to share information across different pedestrians in crowded scenes, we propose to consider the pedestrians in a scene as nodes on the graph and leverage the recent progress in GNNs. Since GAT allows for aggregating information from neighbors by assigning different importance to different nodes, we use GAT as our sharing mechanism. As shown in Fig. 3, we use edges on the graph to represent the existence of human-human interactions. Many recent works have pointed out that when considering the influences of other pedestrians, each pedestrian in the scene is necessary [38, 10, 12, 33]. Following these works, the pedestrians in the scene are treated as nodes on the complete graph at each time-step.

GAT operates on graph-structured data and computes the features of each graph node by attending over its neighbors, following a self-attention strategy. GAT is constructed by stacking graph attention layers. We illustrate a single graph

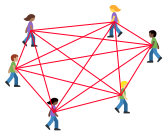

Figure 3. Pedestrians in a scene are considered as nodes on the complete graph at every time-step. The edges on the graph represent the exist of human-human interactions.

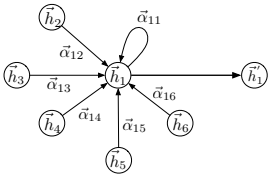

Figure 4. An illustration of graph attention layer. It allows a node to assign different importance to different nodes within a neighborhood and aggregate features from them.

attention layer in Fig. 4. The input of the graph attention layer is $h = \{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_N\}$, where $\vec{h}_i \in R^F$, N is the number of nodes, and F is the feature dimension of each node. The output is $h' = \{\vec{h}'_1, \vec{h}'_2, \ldots, \vec{h}'_N\}$. ($\vec{h}'_i \in R^{F'}$, F' and F can be unequal).

In observation period, m_i^t $(t = 1, \ldots, T_{obs})$ is fed to graph attention layer. The coefficients in the attention mechanism of the node pair (i, j) can be computed by:

$$\alpha_{ij}^t = \frac{\exp(\text{LeakyReLU}(a^T[\mathbf{W}m_i^t \parallel \mathbf{W}m_j^t])}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(a^T[\mathbf{W}m_i^t \parallel \mathbf{W}m_k^t]))} \tag{4}$$

where $\parallel$ is the concatenation operation, $\cdot^T$ represents transposition, α_{ij}^t is the attention coefficient of node j to i at time-step t, $\mathcal{N}_i$ represents the neighbors of node i on the graph. $\mathbf{W} \in R^{F' \times F}$ is the weight matrix of a shared linear transformation which is applied to each node (F is the dimension of m_i^t , F' is the dimension of output), and $a \in R^{2F'}$ is the weight vector of a single-layer feedforward neural network. It is normalized by a softmax function with LeakyReLU.

After getting the normalized attention coefficients, the output of one graph attention layer for node i at t is given by:

$$\hat{m}_i^t = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^t \mathbf{W} m_j^t\right) \tag{5}$$

where σ is a nonlinear function. Eq. 4 and Eq. 5 show how a single graph attention layer works. In our implementation, two graph attention layers are adopted. $\hat{m}_i^t$ (the result after two graph attention layers) is the aggregated hidden state for pedestrian i at t, which contains the spatial influence from other pedestrians.

3.4. Fusion of Spatial and Temporal Information

When modeling interactions in crowded scenes, many LSTM-based methods share hidden states among pedestrians [1, 12, 38]. However, these methods only consider the hidden information at the same time-step. In our work, we propose to use another LSTM to model the temporal correlations between interactions explicitly. We term this LSTM as G-LSTM:

$$g_i^t = \text{G-LSTM}(g_i^{t-1}, \hat{m}_i^t; W_g) \tag{6}$$

where $\hat{m}_i^t$ is from Eq. 5. W_g is the G-LSTM weight and is shared among all the sequences.

In Encoder component, two LSTMs (M-LSTM, G-LSTM) are used to model the motion pattern of each pedestrian, and the temporal correlations of interactions, respectively. We combine these two parts to accomplish the fusion of spatial and temporal information. At time-step T_{obs}, there are two hidden variables ($m_i^{T_{obs}}$, $g_i^{T_{obs}}$) from two LSTMs of each pedestrian. In our implementation, these two variables are fed to two different multilayer perceptrons ($\delta_1(\cdot)$ and $\delta_2(\cdot)$) before getting concatenated:

$$\bar{m}_i = \delta_1(m_i^{T_{obs}}) \tag{7}$$

$$\bar{g}_i = \delta_2(g_i^{T_{obs}}) \tag{8}$$

$$h_i = \bar{m}_i \parallel \bar{g}_i \tag{9}$$

3.5. Future Trajectory Prediction

From the real-world trajectory datasets, we need to learn human motion patterns, which is the response of pedestrians to the changing environment.. Due to the uncertainty of pedestrian movement, we hope our model can generate multiple reasonable and realistic trajectories.

Most previous works [1, 33, 13] embody this uncertainty by learning parameters of Gaussian distribution, then obtain future positions sampled from the distribution. During the training phase, these models minimize the negative log-likelihood loss of ground-truth positions under the predicted Gaussian distribution. However, such methods introduce difficulty in backpropagation as the sampling process is non-differentiable [12]. Gupta *et al.* [12] proposed a variety loss to encourage the network to produce diverse samples, and verified the effectiveness of their method. We follow their strategy to model the multimodal properties of pedestrian motion.

The intermediate state vector of our model consists of three parts: hidden variables of M-LSTM, hidden variables of G-LSTM, and the noise added (as shown in Fig. 2). The intermediate state vector is calculated as:

$$d_i^{T_{obs}} = h_i \parallel z \tag{10}$$

where z represents noise, h_i is from Eq. 9. The intermediate state vector $d_i^{T_{obs}}$ then acts as the initial hidden state of the decoder LSTM (termed as D-LSTM). The predicted relative position is given by:

$$d_i^{T_{obs}+1} = \text{D-LSTM}(d_i^{T_{obs}}, e_i^{T_{obs}}; W_d) \tag{11}$$

$$(\Delta x_i^{T_{obs}+1}, \Delta y_i^{T_{obs}+1}) = \delta_3(d_i^{T_{obs}+1}) \tag{12}$$

where W_d is D-LSTM weight, $\delta_3(\cdot)$ is a linear layer, $e_i^{T_{obs}}$ is from Eq. 2. After getting the predicted relative position at time-step $T_{obs}+1$, the subsequent inputs of D-LSTM are calculated based on the last predicted relative position according to Eq. 2. And it's easy to convert relative positions to absolute positions for computing loss.

The variety loss from [12] works as follow: for each pedestrian, the model produces multiple predicted trajectories by randomly sampling z from $\mathcal{N}(0,1)$ (the standard normal distribution). Then it chooses the trajectory that has the smallest distance to ground-truth as the model output to compute the loss:

$$L_{variety} = \min_k ||Y_i - \hat{Y}_i^k||_2 \tag{13}$$

where Y_i is the ground-truth trajectory of pedestrian i, $\hat{Y}_i^k$ is the trajectory produced by our model, k is a hyperparameter. By considering only the best trajectory, this loss encourages the network to cover the space of outputs that conform to the past trajectory.

3.6. Implementation Details

All LSTMs in our implementation only have one layer. The dimension of e_i^t in Eq. 2 is set to 16; the dimension of m_i^t in Eq. 3 is set to 32. $\mathbf{W}$ in Eq. 4 for first graph attention layer is of shape 16×16, for the second layer is of shape 16×32, the dimension of a in Eq. 4 is set to 32 for first graph attention layer, and set to 64 for second layer. Batch Normalization is applied over the input of graph attention layer. The dimension of g_i^t in Eq. 6 is set to 32. $\delta_1(\cdot)$ in Eq. 7 contains 3 layers with ReLU activation functions, where the hidden nodes in these layers are 32, 64 and 24, respectively. $\delta_2(\cdot)$ in Eq. 8 contains 3 layers with ReLU activation functions and the number of hidden nodes is 32, 64, 16, respectively. The dimension of z in Eq. 10 is set to 16. We train the network using Adam optimizer with a learning rate of 0.01 and batch size 64.

4. Experiments

In this section, we present the results on two publicly available pedestrian trajectory datasets: ETH [21] and UCY [17]. The ETH dataset consists of two scenarios named ETH and HOTEL. UCY dataset includes two scenarios and in three components, named ZARA-01, ZARA-02 and UCY. These datasets contain thousands of real-world pedestrian trajectories and cover rich human-human interactions. Because the original ETH and UCY datasets do not have a uniform data format, we follow the same data preprocessing strategy as SGAN [12] to compare different methods. First, all the data is converted to the world coordinate system and then interpolated to obtain values at every 0.4 seconds.

Evaluation Metrics: same as prior works [33, 12, 38], we use two error metrics to report prediction errors:

1. *Average Displacement Error (ADE)*: the mean square error (MSE) over all estimated positions in the predicted trajectory and ground-truth trajectory.

2. *Final Displacement Error (FDE)*: the distance between the predicted final destination and the true final destination at T_{pred}.

Baselines. Since traditional methods based on handcrafted features (such as linear model, social force model and interacting gaussian processes, etc.) perform worse than social LSTM model [1], we compare our model with following methods:

1. ***LSTM***: a vanilla LSTM with no pooling mechanism, all trajectories are considered to be independent of each other.

2. ***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.

3. ***Social Attention***: the method proposed by Vemula *et al*. [33]. It formulates the trajectory prediction problem as a spatial-temporal graph, and uses two types of edges to capture the spatial and temporal dynamics of the crowd.

4. ***CIDNN***: the method proposed by Xu *et al*. [38]. It has four components: motion encoder module, location encoder module, crowd interaction module, and displacement prediction module.

5. ***SGAN***: the method proposed by Gupta *et al*. [12]. We represent results of the model with three different control settings: *SGAN-1V-1*, *SGAN-20V-20* and *SGAN-20VP-20* (the meaning of these notations will be explained later).

For ablation study, we investigate our model with different control settings, following a similar notation as [12]. We represent our full method as *STGAT-kV-N*, k represents the number of outputs for calculating the variety loss (as shown in Eq. 13), and $k=1$ means no variety loss, N refers to the number of sampling times during test time (the definitions of k and N in our model are the same with k and N in SGAN model, p in SGAN represents the usage of the "pooling module"). And we use the best prediction in L_2 sense for quantitative evaluation.

In addition to the models with different control settings, we investigate a variation of STGAT to capture the contributions of different parts of our model. In this case, we ignore the temporal correlations of interactions and only use

Metric	Dataset	LSTM	S-LSTM [1]	SocialAttention [33]	CIDNN [38]	SGAN [12]			STGAT(Ours)			
						1V-1	20V-20	20VP-20	1V-1	1V-20	20V-20	SGAT
ADE	**ETH**	0.70 / 1.09	0.73 / 1.09	1.04 / 1.39	0.89 / 1.25	0.79 / 1.13	0.61 / 0.81	0.60 / 0.87	0.75 / 0.88	0.74 / 0.80	**0.56 / 0.65**	0.68 / 0.70
	HOTEL	0.55 / 0.86	0.49 / 0.79	1.95 / 2.51	1.25 / 1.31	0.71 / 1.01	0.48 / 0.72	0.52 / 0.67	0.43 / 0.56	0.42 / 0.52	**0.27 / 0.35**	0.32 / 0.37
	UNIV	0.36 / 0.61	0.41 / 0.67	0.78 / 1.25	0.59 / 0.90	0.37 / 0.60	0.36 / 0.60	0.44 / 0.76	0.31 / 0.52	**0.31 / 0.51**	0.32 / 0.52	0.35 / 0.59
	ZARA1	0.25 / 0.41	0.27 / 0.47	0.59 / 1.01	0.29 / 0.50	0.25 / 0.42	**0.21 / 0.34**	0.22 / 0.35	0.25 / 0.41	0.24 / 0.39	**0.21 / 0.34**	0.21 / 0.35
	ZARA2	0.31 / 0.52	0.33 / 0.56	0.55 / 0.88	0.28 / 0.51	0.32 / 0.52	0.27 / 0.42	0.29 / 0.42	0.21 / 0.31	0.20 / 0.30	**0.20 / 0.29**	0.24 / 0.31
AVG		0.43 / 0.70	0.45 / 0.72	0.98 / 1.41	0.66 / 0.89	0.49 / 0.74	0.39 / 0.58	0.41 / 0.61	0.39 / 0.54	0.38 / 0.50	**0.31 / 0.43**	0.36 / 0.47
FDE	**ETH**	1.45 / 2.41	1.48 / 2.35	1.83 / 2.39	1.89 / 2.32	1.61 / 2.21	1.22 / 1.52	1.19 / 1.62	1.55 / 1.66	1.52 / 1.53	**1.10 / 1.12**	1.29 / 1.35
	HOTEL	1.17 / 1.91	1.01 / 1.76	2.97 / 2.91	2.20 / 2.36	1.44 / 2.18	0.95 / 1.61	1.02 / 1.37	0.88 / 1.15	0.85 / 1.08	**0.50 / 0.66**	0.56 / 0.67
	UNIV	0.77 / 1.31	0.84 / 1.40	1.56 / 2.54	1.13 / 1.86	0.75 / 1.28	0.75 / 1.26	0.84 / 1.52	0.66 / 1.13	0.65 / 1.12	**0.66 / 1.10**	0.73 / 1.23
	ZARA1	0.53 / 0.88	0.56 / 1.00	1.24 / 2.17	0.59 / 1.04	0.53 / 0.91	0.42 / 0.69	0.43 / 0.68	0.53 / 0.91	0.50 / 0.87	**0.42 / 0.69**	0.41 / 0.69
	ZARA2	0.65 / 1.11	0.70 / 1.17	1.09 / 1.75	0.60 / 1.07	0.66 / 1.11	0.54 / 0.84	0.58 / 0.84	0.44 / 0.68	0.42 / 0.64	**0.40 / 0.60**	0.46 / 0.64
AVG		0.91 / 1.52	0.91 / 1.54	1.74 / 2.35	1.28 / 1.73	1.00 / 1.54	0.78 / 1.18	0.81 / 1.21	0.81 / 1.11	0.79 / 1.05	**0.62 / 0.83**	0.69 / 0.92

Table 1. Quantitative results of all the baselines and our model with different control settings. We represent two error metrics ADE and FDE for $T_{pred} = 8$ and $T_{pred} = 12$. The experiments show that our proposed model significantly improves the prediction accuracy compared to other baseline methods (lower numerical results are better). STGAT-1V-1, STGAT-1V-20 and STGAT-20V-20 are models with different control settings, and SGAT is a variation of our model.

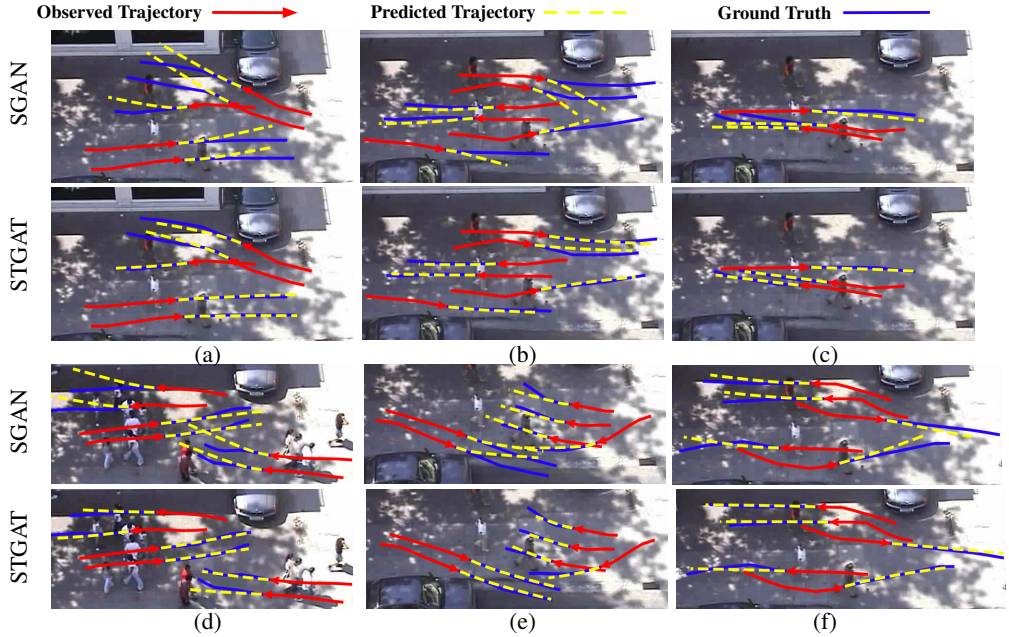

Figure 5. Comparisons between our model (STGAT-20V-20) with SGAN (SGAN-20VP-20) in six different scenarios. We choose scenarios with multiple pedestrians and complex interactions. For a better view, only part of the pedestrians in the scene is presented. We can see that our model can generate complex trajectories, while SGAN produces more linear trajectories. And our trajectories are closer to ground-truth.

LSTM once for each pedestrian in Encoder. The hidden state of the last observation time-step is processed by GAT (GAT is used only once for each pedestrian). The rest of the model (including the variety loss, the noise, etc.) is the same as STGAT-20V-20. We refer to this variation as **SGAT** (i.e. STGAT model without considering temporal correlations of interactions).

Evaluation Methodology. Following prior works [12, 1, 38], the leave-one-out approach is adopted. The model is trained on 4 datasets and tested on the remaining dataset. We observe a trajectory for 3.2 seconds (8 time-steps), then predict for the next 3.2 seconds (8 time-steps) and 4.8 seconds (12 time-steps).

4.1. Quantitative Evaluation

Following SGAN [12], we refer to our complete method as STGAT-20V-20. In Table 1, we evaluate our model against all baseline models as well as our model with multiple control settings. The results show that our method outperforms all compared methods on all datasets in terms of ADE and FDE. The best baseline method that has the lowest average prediction error is SGAN-20V-20. Compared to it, the average error rate of our method in ADE is reduced by 25.8% and 34.9% respectively, when predicting the future 8, 12 time-steps. For FDE, the performance is increased by 25.8%, 42.2% respectively. These results show that our model has advantages compared to other methods, especially in the case of longer predictions ($T_{pred} = 12$).

Evaluation of GAT. The SGAT model only uses one LSTM for each pedestrian, and GAT is adopted at T_{obs}. Its architecture is similar to SGAN-20VP-20 [12], but GAT is exploited to aggregate information from others. The results of SGAT show the ability of GAT for modeling inter-

	SGAN-P	SocialAttention	SGAT	STGAT
8	8.910	29.328	12.268	12.502
12	8.247	27.804	10.861	10.858
AVG	8.579	28.566	11.565	11.680
Time Usage	1x	3.33x	1.35x	1.36x

Table 2. The comparison of speed in seconds. All methods are benchmarked on the same dataset (containing 2875, 2253 trajectories for two prediction lengths) and one NVIDIA TITAN Xp graphics card.

actions among pedestrians. As is shown in Table 1, compared with SGAN-20VP-20, for two prediction lengths, the average error rate of SGAT in ADE is reduced by 13.9% and 29.8% respectively, and in FDE is reduced by 17.4% and 31.5% respectively. Compared with vanilla LSTM, the performance in ADE is increased by 19.4% and 48.9% respectively. While for FDE, the performance is increased by 31.8% and 65.2% respectively. These results validate the effectiveness of the GAT component.

Evaluation of G-LSTM. The SGAT can be seen as a simplified version of our full model. The only difference is that STGAT has G-LSTM after GAT module. As shown in Table 1, the STGAT-20V-20 model outperforms the SGAT model on average. Specifically, the average error rate of STGAT-20V-20 in terms of ADE is reduced by 16.1% and 9.3% respectively. In terms of FDE, the average error rate is reduced by 11.3%, 10.8% respectively. Obviously, modeling the temporal correlations of interactions contributes to improving the performance.

Evaluation of variety loss. Due to the polymorphism of pedestrian movement, we use the variety loss [12] to produce multiple socially acceptable trajectories. We represent three different control settings of our model in Table 1. Both STGAT-1V-20 and STGAT-20V-20 can generate multiple future trajectories. By using the variety loss, the average error rate of STGAT-20V-20 in ADE is reduced by 22.6% and 16.3% respectively. For FDE, the average error rate is reduced by 27.4%, 26.5% respectively.

Time and space consumption. We compare our method with two baselines, SGAN [12] and SocialAttention [33][1]. As shown in Table 2, STGAT is slower than SGAN. This is caused by our "GAT" scheme which is more time consuming than the "Pooling" module of SGAN. Table 4 compares the CUDA memory usage of each model during the training and evaluating phases. The memory usage of STGAT is 2.5 times larger than SGAN during training. The comparison between SGAT and STGAT indicates that considering the continuity of interactions will not influence the time consuming, but significantly increase the memory usage.

[1]The consumption of time and space is greatly affected by the implementation details. We use the official implementation of the corresponding models. Note that SocialAttention model is implemented with a batch size of 1 during evaluating in original code.

	SGAN-P	SocialAttention	SGAT	STGAT
train	3031	649	1133	7503
evaluation	589	657	589	593

Table 3. The comparison of CUDA memory usage (in MB). SGAN-P, SGAT, and STGAT are benchmarked with the same batch size (32) and prediction length (12).

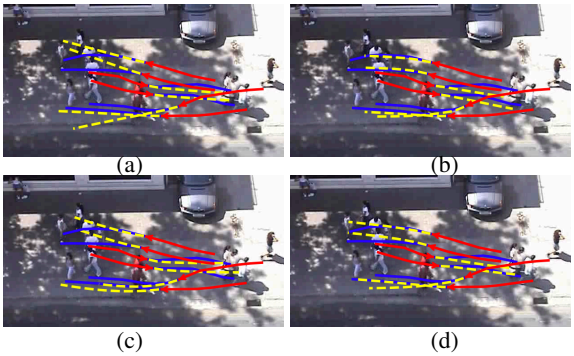

Figure 6. Example of diverse predictions. (a) shows the best sample produced by SGAN (SGAN-20VP-20) model. (b)(c)(d) are three diverse samples generated by our STGAT-20V-20 model, where (d) represents the best sample from our model.

Figure 7. Trajectories generated by SGAT model. (a) shows the same scenario as Fig. 5(b). (b) shows the same scenario as Fig. 5(e).

4.2. Qualitative Evaluation

As mentioned before, pedestrian trajectory prediction is a complex problem because we have to consider the spatial-temporal properties of each pedestrian in the scene. Pedestrians in crowded scenes may have complex interactions, representing different motion modes, including forming groups, following other pedestrians, changing directions to avoid collisions, etc. The qualitative results are shown in Fig. 5. We choose scenarios containing different motion patterns and collision avoidance. Fig. 5 shows that SGAN (SGAN-20VP-20) model can capture interactions, and generate socially-acceptable trajectories in most cases. However, compared with the trajectories produced by STGAT, their trajectories are closer to linear. And our model outperforms SGAN as we generate trajectories closer to ground-truth, especially when the crowd moves in the opposite direction.

Fig. 6 shows an example of diverse predictions. We represent a challenging scenario with multiple pedestrians and complex interactions. Fig. 6(b)(c)(d) show three different predictions generated by our model, where (d) represents the sample closest to the ground-truth (we term this prediction the best prediction). As a comparison, we show the

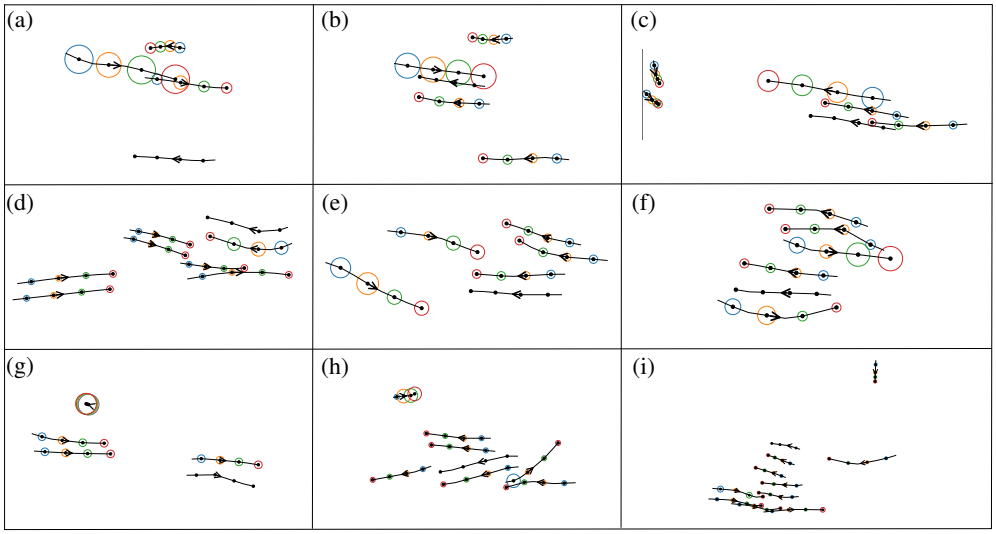

Figure 8. Attention weights predicted by graph attention mechanism. The solid dots on the trajectory indicate different time-steps and the arrows show the directions of trajectories. The trajectories without circles are the target pedestrians. The circles on other trajectories show the attentions represented with the radius proportional to attention weight.

best prediction produced by SGAN in Fig. 6(a). As can be seen from Fig. 6(a) and Fig. 6(d), neither model can predict accurate future trajectories in such a complex environment.

Fig. 7 presented the predicted trajectories of the SGAT model in two scenarios. By comparing the corresponding scenes of Fig. 5, it can be observed that the SGAT model is worse than the STGAT model, which is consistent with the quantitative results. By comparing Fig. 7(a) with Fig. 5(b), Fig. 7(b) with Fig. 5(e), we can see that the trajectories produced by SGAT model are socially plausible, but the accuracy is worse than STGAT model. These qualitative results visually demonstrate that when considering the temporal correlations of interactions, the predicted trajectories are more accurate and socially acceptable.

We visualize the learned attention weights in Fig 8. As shown in Fig. 8(a)-(e), our model successfully learned the relative importance of surrounding pedestrians in these scenarios. In these successful cases, GAT assigns higher weights to some neighbours such as moving in the opposite directions and being close in position. In addition, when surroundings move toward the same direction, the pedestrians in front has more significant influence than pedestrians in the rear. And the model assigns nearly equal attention weights to pedestrians who are far from the target. Since the input to our model is the relative displacement of each pedestrian relative to the previous moment, these learned weights are based on each pedestrian's motion state. In this process, neither global nor local position information is adopted. These successful cases show that GAT can assign reasonable importance to neighbours by their motion status.

There are also many failure cases as shown in Fig. 8 (f)-(i). In Fig. 8(f), only part of the weights is reasonable. In Fig. 8(g), the stationary pedestrian has an unreasonable high influence. And in Fig. 8(h)(i), the learned attention weights are very chaotic. In these failed cases, (g) and (i) are very representative. Our model often assigns a high impact on the stationary pedestrian, and when the scene contains many pedestrians, the weights assigned are confusing. The cause of the first problem may be that we use relative displacement as the model input. The possible reason for the second problem is that all pedestrians in the scene are considered. We will solve these problems in future work.

5. Conclusion

In this work, a novel seq2seq framework that can jointly predict the future trajectories of all pedestrians in a scene is proposed. We use one LSTM for each trajectory to capture the historical trajectory information of each pedestrian, and adopt graph attention network to model the interactions in human crowds at every time-step. Moreover, another LSTM is adopted to model the temporal correlations between interactions explicitly. Our proposed method outperforms state-of-the-art methods on two publicly available datasets. Qualitative experiments show that graph attention network can assign reasonable importance to neighbors according to their motion states, and our model can predict accurate trajectories in different scenes.

6. Acknowledgements

This work is in part supported by the National Key Research and Development Program of China (2017YFC0804900, 2017YFB1002600), the National Natural Science Foundation of China (61532002, 61702482), the 13th Five-Year Common Technology pre Research Program (41402050301-170441402065), the Science and Technology Mobilization Program of Dongguan (KZ2017-06).

References

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

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

[3] Gianluca Antonini, Michel Bierlaire, and Mats Weber. Discrete choice models of pedestrian walking behavior. *Transportation Research Part B: Methodological*, 40(8):667–687, 2006.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[5] Aniket Bera, Sujeong Kim, Tanmay Randhavane, Srihari Pratapa, and Dinesh Manocha. Glmp-realtime pedestrian path prediction using global and local movement patterns. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5528–5535. IEEE, 2016.

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[7] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. End-to-end continuous speech recognition using attention-based recurrent nn: First results. *arXiv preprint arXiv:1412.1602*, 2014.

[8] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[9] Rémi Emonet, Jagannadan Varadarajan, and Jean-Marc Odobez. Extracting and locating temporal motifs in video scenes using a hierarchical non parametric bayesian model. In *CVPR 2011*, pages 3233–3240. IEEE, 2011.

[10] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. *Neural networks*, 108:466–478, 2018.

[11] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning*, pages 1764–1772, 2014.

[12] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018.

[13] Irtiza Hasan, Francesco Setti, Theodore Tsesmelis, Alessio Del Bue, Fabio Galasso, and Marco 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*, pages 6067–6076, 2018.

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

[15] Timothy M Hospedales, Jian Li, Shaogang Gong, and Tao Xiang. Identifying rare and subtle behaviors: A weakly supervised joint topic model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2451–2464, 2011.

[16] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 336–345, 2017.

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

[18] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[19] Jim Mainprice, Rafi Hayne, and Dmitry Berenson. Goal set inverse optimal control and iterative replanning for predicting human reaching motions in shared workspaces. *IEEE Transactions on Robotics*, 32(4):897–908, 2016.

[20] Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 935–942. IEEE, 2009.

[21] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268. IEEE, 2009.

[22] Rohit Prabhavalkar, Kanishka Rao, Tara N Sainath, Bo Li, Leif Johnson, and Navdeep Jaitly. A comparison of sequence-to-sequence models for speech recognition. In *Interspeech*, pages 939–943, 2017.

[23] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016.

[24] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 300–311, 2017.

[25] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[26] Tianmin Shu, Sinisa Todorovic, and Song-Chun Zhu. Cern: confidence-energy recurrent network for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5523–5531, 2017.

[27] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm

network for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1227–1236, 2019.

[28] Hang Su, Yinpeng Dong, Jun Zhu, Haibin Ling, and Bo Zhang. Crowd scene understanding with coherent recurrent neural networks. In *IJCAI*, volume 1, page 2, 2016.

[29] Hang Su, Jun Zhu, Yinpeng Dong, and Bo Zhang. Forecast the plausible paths in crowd scenes. In *IJCAI*, volume 1, page 2, 2017.

[30] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[31] Adrien Treuille, Seth Cooper, and Zoran Popović. Continuum crowds. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 1160–1168. ACM, 2006.

[32] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[33] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7. IEEE, 2018.

[34] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015.

[35] Xiaogang Wang, Xiaoxu Ma, and W Eric L Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Transactions on pattern analysis and machine intelligence*, 31(3):539–555, 2008.

[36] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[37] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[38] Yanyu Xu, Zhixin Piao, and Shenghua Gao. Encoding crowd interaction with deep neural network for pedestrian trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5275–5284, 2018.

[39] Kota Yamaguchi, Alexander C Berg, Luis E Ortiz, and Tamara L Berg. Who are you with and where are you going? In *CVPR 2011*, pages 1345–1352. IEEE, 2011.

[40] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[41] Shuai Yi, Hongsheng Li, and Xiaogang Wang. Pedestrian behavior understanding and prediction with deep neural networks. In *European Conference on Computer Vision*, pages 263–279. Springer, 2016.

[42] Bolei Zhou, Xiaogang Wang, and Xiaoou Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2871–2878. IEEE, 2012.

[43] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.