



Dual-branch spatio-temporal graph neural networks for pedestrian trajectory prediction

Xingchen Zhang^{a,*}, Panagiotis Angeloudis^b, Yiannis Demiris^a

^a Personal Robotics Laboratory, Department of Electrical and Electronic Engineering, Imperial College London, South Kensington Campus, Exhibition Road, London, SW7 2AZ, UK

^b Transport Systems and Logistics Laboratory, Department of Civil and Environmental Engineering, Imperial College London, South Kensington Campus, Exhibition Road, London, SW7 2AZ, UK

ARTICLE INFO

Article history:

Received 29 April 2022

Revised 7 April 2023

Accepted 21 April 2023

Available online 28 April 2023

Keywords:

Pedestrian trajectory prediction

Social interactions

Graph convolutional networks

Graph attention networks

Spatio-temporal graph

ABSTRACT

Pedestrian trajectory prediction is an important area in computer vision, with wide applications in autonomous driving, robot path planning, and surveillance systems. The core underlying technique of these applications is pattern recognition. A key challenge in this area is modeling social interactions between pedestrians, such as pedestrian view area and group behaviors. However, although many methods have been proposed to model social interactions, pedestrian view area and group behaviors have not been explored together to account for complex situations. Additionally, most existing studies require additional detectors and manual annotations to handle view area and group interactions, respectively. In this paper, we propose a dual-branch spatio-temporal graph neural network to automatically model view area and grouping together. Specifically, a spatio-temporal graph attention network (STGAT) branch is designed to handle pedestrian view area, and a spatio-temporal graph convolutional network (STGCN) branch is designed to model group interactions. The features of these branches are then fused to provide better feature representations, on which a temporal convolution operation (TCN) is performed for trajectory prediction. Experiments on public standard datasets demonstrate that the proposed method achieves very competitive performance and predicts socially acceptable trajectories in different challenging scenarios.

© 2023 The Authors. Published by Elsevier Ltd.

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

1. Introduction

Pedestrian trajectory prediction has attracted much attention due to its applications in various areas, such as autonomous driving [1–3], robot navigation in crowds [4,5], and surveillance systems [6–8]. Although considerable progress has been made in this field, it remains a challenging problem.

Apart from the effect of static objects in the environment, a key task in trajectory prediction is the modeling of social interactions between pedestrians. Over the past years, many nice methods have been proposed to model social interactions. For example, the Social Force method [9] modeled social interactions using attractive and repulsive forces. Social-LSTM [10] utilized a pooling mechanism to aggregate the information of a pedestrian and his/her neighbors. Social GAN [11] applied max pooling to all pedestrians to consider long-range interactions. Zamboni et al. [12] employed convolutional neural networks to model social

interactions. Furthermore, graph-based methods have been proposed to predict pedestrian trajectories in recent years by considering pedestrians as nodes and social interactions as edges between nodes. For example, Kosaraju et al. [13] proposed the Social-BiGAT that combined generative adversarial network (GAN) and graph attention network (GAT) to learn embedding for interactions. Huang et al. [14] proposed the STGAT to model interactions using GAT together with Long-Short Term Memory (LSTM) units. Mohamed et al. proposed to handle pedestrian trajectory prediction using spatio-temporal graph convolutional networks (GCNs) [15]. Hu et al. [16] proposed the neural motion message passing (NMMP) in a graph to model interactions.

Although promising results have been achieved, existing methods have some limitations when modeling social interactions. First, all pedestrians are considered in most graph-based methods [13–16], as shown in Fig. 1(a) and (b). However, the motion of a pedestrian is usually only affected by pedestrians inside his/her view

* Corresponding author.

E-mail address: xingchen.zhang@imperial.ac.uk (X. Zhang).

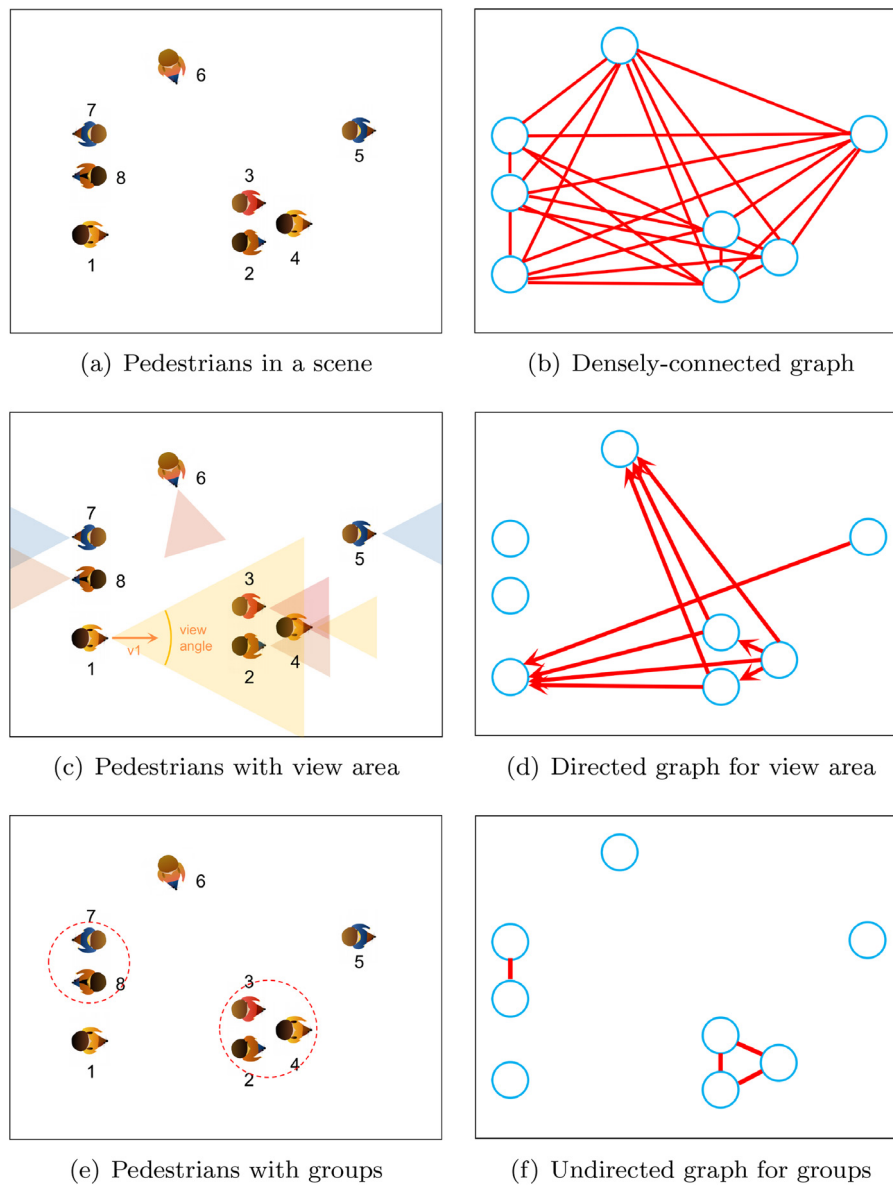


Fig. 1. Pedestrians in a scene (the first column) and corresponding graphs (the second column). (a) Pedestrians without considering view area and group information. (b) Densely-connected graph of the scene (a) used by most graph-based methods. (c) Pedestrians with view area. Take pedestrian 1 as an example, only pedestrians 2, 3, 4, and 5 affect his/her path at this time instant. (d) Directed graph modeling the view area of pedestrians in this work. (e) Pedestrians with group information. (f) Undirected graph modeling group information in this work. We create both directed and undirected graphs to encode view area and group information simultaneously.

area¹. For example, as shown in Fig. 1(c), pedestrian 2 will affect the motion of pedestrian 1, but pedestrian 1 will not affect pedestrian 2. In other words, the effects of view area between pedestrians are asymmetric. Therefore, only a part of pedestrians is needed in computing representations of pedestrians (Fig. 1(d)). Some methods [13,14] can learn to set lower attention weights to pedestrians outside of view areas. However, considering all pedestrians in the scene does not match the actual walking pattern and introduces noises. Therefore, the performance is limited to some extent. Second, it is frequent that pedestrians walk in groups, as shown in Fig. 1(e). Pedestrians in the same group tend to move in a similar pattern over a certain period. Therefore, incorporating group information is imperative for predicting better trajectories. However, existing methods [14–17] have rarely consid-

ered these group interactions. A few studies [18,19] have tried to model group behavior. However, manual annotation of groups is required. Third, these two kinds of social interactions are both crucial, thus they should be considered together to predict better trajectories. However, to the best of our knowledge, no methods have considered both view area and grouping simultaneously.

To solve these issues, we propose a dual-branch spatio-temporal graph neural network (Social TAG) to model complex interactions between pedestrians in this paper. Specifically, we formulate social interactions as view area (asymmetric interactions) and grouping (symmetric interactions). A spatio-temporal graph attention network (STGAT) branch with a Finding View Area module is designed to model pedestrian view area (Fig. 1(d)), and a spatio-temporal graph convolutional network (STGCN) branch with a Finding Group module is designed to capture group behaviors automatically (Fig. 1(f)). The features provided by the two branches are fused and sent to a temporal convolution operation (TCN) to predict future trajectories.

¹ Environment (context) is also important. However, it is not the focus of this paper, thus we leave it for our future work

Our method has significant differences compared with existing studies. First, we consider both view area and grouping, which are both crucial in pedestrian trajectory prediction. Second, we find view area based on pedestrian walking direction automatically without using additional detectors. By contrast, the MX-LSTM [20,21] built view areas based on head pose and required an additional head pose detector. In addition, pedestrian head pose is not always consistent with walking direction because pedestrians may look around when they are walking. In this case, performing collision avoidance based on head pose may not be accurate. Third, our method finds groups automatically based on the position and walking direction of pedestrians, while both CoMoGCN [18] and RSBG [19] require manual annotations of groups. The only input of the proposed method is pedestrian locations at different time steps. In other words, no additional head pose detector and group annotations are needed. Consequently, the proposed method is effective in capturing complex social interactions and is easy to use.

In summary, the main contributions of this work include:

- We propose a dual-branch spatio-temporal graph-based network to model complex social interactions in pedestrian trajectory prediction. To the best of our knowledge, this is the first attempt to utilize GAT and GCN together to fully model complex social interactions in pedestrian trajectory prediction.
- Unlike existing methods that consider all pedestrians in the scene, we design a realistic Finding View Area module to automatically find pedestrians in the view area of each pedestrian in dynamic scenes. We encode this information in asymmetric adjacency matrices of the STGAT network to model pedestrian view areas.
- Unlike existing methods that require manual annotations of groups, we design a Finding Group module to find groups automatically, which are then encoded in symmetric adjacency matrices of the STGCN network to model pedestrian group behaviors.
- Experiments on public standard datasets demonstrate that the proposed Social TAG achieves very competitive performance and can predict socially acceptable trajectories in various social norms and challenging scenarios.

2. Related work

2.1. Pedestrian trajectory prediction

Pedestrian trajectory prediction has been investigated for many years, including prediction from surveillance view [14–16] and egocentric view [2,3]. Trajectory prediction from surveillance view has wide applications in surveillance, while trajectory prediction from egocentric view can be applied to autonomous vehicles [2,3], robot navigation [4,5], etc. In this paper, we focus on the pedestrian trajectory prediction task from a surveillance view. The main challenges of this task include social interactions between pedestrians, the effect of objects in the environment, and the multi-modal nature of this problem [11]. Some methods have been proposed to handle the effect of the environment [22–24] and the multi-modal nature of trajectory prediction [10,25], while most methods are about the modeling of social interactions.

Common social interaction modeling methods can be generally classified into three types [16]. The first type is conventional methods based on hand-crafted features. For instance, the Social Force method [9] models social interactions using attractive and repulsive forces. However, these conventional methods cannot generalize to complex scenes. The second type is non-graph deep learning-based methods. For example, Social LSTM [10] is one of the earliest deep learning-based pedestrian prediction methods. It utilizes a recurrent network (RNN) to model the motion of pedestrians

and then uses a pooling mechanism to aggregate the outputs of RNNs to generate features. Next [26] and SR-LSTM [27] improved Social LSTM by using more visual features and new pooling mechanisms. To alleviate the issue of LSTM, i.e., sequential processing, transformers have been recently applied in trajectory prediction [28]. Another interesting idea is to first infer the endpoint distribution and then use the sampled endpoints to condition the trajectory prediction process [29]. The third type is graph neural networks (GNNs)-based methods [13–15]. In these methods, pedestrians are treated as nodes of a graph, and interactions are considered as edges between nodes. For example, Huang et al. [14] proposed the STGAT that modeled the social interactions between pedestrians using spatio-temporal GAT. Mohamed et al. [15] proposed the Social-STGCN to model the social interactions of pedestrians using a spatio-temporal GCN. However, both STGAT and Social-STGCN considered all pedestrians, which does not match the fact that each pedestrian has a limited field of view. In addition, both STGAT and Social-STGCN did not consider group behaviors.

For view area modeling, MX-LSTM [20,21] considered pedestrian view area, which is similar to our Finding View Area module. However, it built the view area based on head pose and required an additional head pose detector. Besides, it divided the view area into different cells and then applied pooling, which was not as effective as our GAT-based implementation. In addition, it only modeled the view area. Zhang et al. [30] also only considered the view area.

Regarding grouping modeling, a few studies have been performed. For example, a recent graph-based method (CoMoGCN) proposed by Chen et al. [18] considered group information. However, offline preprocessing was needed to obtain group labels. Another method (RSBG) proposed by Sun et al. [19] also considered group behaviors. However, manual annotations of groups were required for training, which was labor-intensive and not automatic. In addition, the group defined in RSBG contained pedestrians walking together and included those who have similar destinations. However, the latter is difficult to identify, especially in long-time dynamic videos. Moreover, Zhou et al. [31] proposed a method (SGC-LSTM) based on GNN and LSTM to predict pedestrian trajectories. Specifically, a companion loss was designed to encourage the model to consider the overall similarity of pedestrian trajectories in each group. They considered pedestrians to be in a group if they walked close to each other. However, all these three methods did not consider the pedestrian view area.

In summary, existing methods either neglect view area (asymmetric interactions) [18,19,31] or group behaviors (symmetric interactions) of pedestrians [20,21], or they do not consider both [14,15,32]. The proposed Social TAG solves these issues by automatically modeling both pedestrian view area and group behaviors. The only input is pedestrian locations at different time steps, i.e., no additional head pose detector and group annotations are needed. Consequently, the proposed method effectively captures complex social interactions and is easy to use.

2.2. Graph neural networks

Graph neural networks (GNNs) are neural networks that apply to graph data [33–35]. The adjacency matrix is used to describe node connections in GNN, where both directed [36] and undirected graphs [15] have been utilized. In an undirected graph, the influence on each other between two nodes are the same, thus the adjacency matrix of an undirected graph is symmetric. In contrast, in a directed graph, the effects between two nodes are not the same, thus the adjacency matrix is an asymmetric matrix. GCN [37] and GAT [38] are two important types of GNN. Specifically, GCN applies convolution operations on graph data, while GAT utilizes the attention mechanism to assign different weights to different nodes via

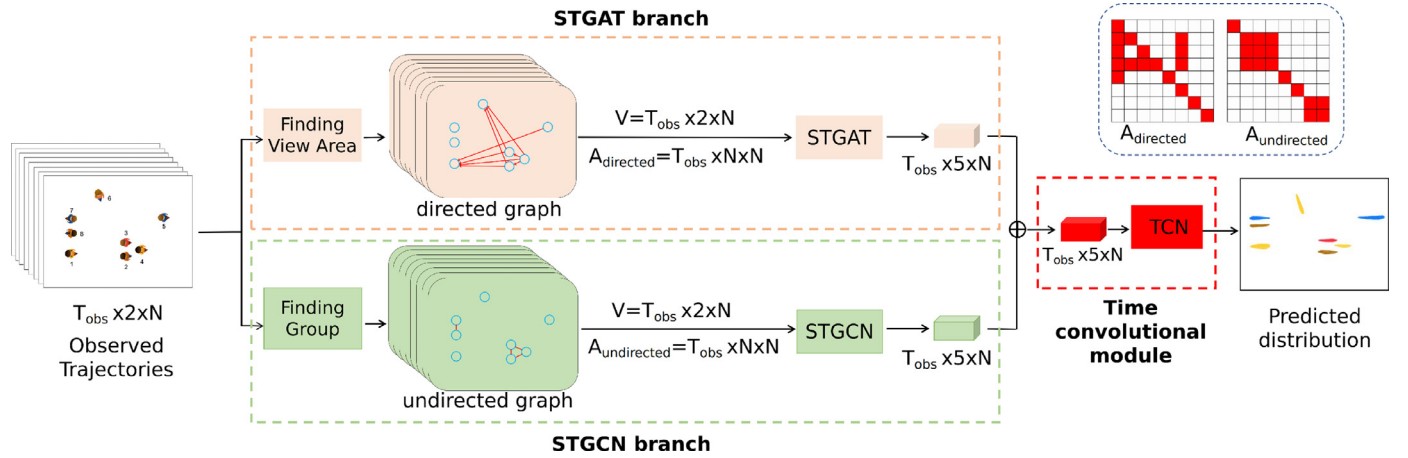


Fig. 2. Overview of the proposed Social TAG method. Social TAG consists of an STGAT branch, an STGCN branch, and a time convolutional module. In the STGAT branch, the view area of each pedestrian and other pedestrians inside the view area are first worked out, based on which directed spatio-temporal graphs are built. STGAT network is then applied to these graphs to obtain STGAT embedding. In the STGCN branch, group information is first worked out, based on which undirected spatio-temporal graphs are created. STGCN network is then applied to these graphs to obtain STGCN embedding. The time convolutional module performs prediction based on the fused embedding. At the right corner are examples of the directed adjacency matrix and undirected adjacency matrix corresponding to Fig. 1(d) and Fig. 1(f). Details of the Finding View Area module can be found in Section 3.3 and details of the Finding Group module can be found in Section 3.4.

learning. The Laplacian matrix in GCN should be symmetric, therefore GCN is usually used to handle undirected graphs. GCN has been widely applied to many applications, such as pedestrian prediction [15,18] and robot navigation in crowds [39]. In contrast, because GAT can assign different attention weights to different nodes, it is usually used to handle directed graphs [13,14].

3. Method

3.1. Problem formulation

The problem that we try to solve in this paper can be formulated as follows: given the past spatial location of N pedestrians in a scene, (x_t^i, y_t^i) , where $i = 1, 2, \dots, N$ and $t = 1, 2, \dots, T_{obs}$, we aim to predict the future positions of these N pedestrians from time steps $T_{obs} + 1$ to T_{pred} . Following most studies in this field [10,15,18], we consider a fixed N in each observed sequence, and each of these N pedestrians is identified and tracked throughout the sequence. In other words, we do not consider that pedestrians may go out or in during the T_{obs} time steps. Considering the multi-modal nature of the trajectory prediction problem, we assume that the trajectory follows bivariate Gaussian distribution following Social LSTM [10] and Social-STGCNN [15]. Therefore, the predicted coordinates $(\hat{x}_t^i, \hat{y}_t^i)$ are

$$(\hat{x}_t^i, \hat{y}_t^i) \sim \mathcal{N}(\mu_t^i, \sigma_t^i, \rho_t^i), \quad (1)$$

where the mean $\mu_t^i = (\mu_{x,t}^i, \mu_{y,t}^i)$, standard deviation $\sigma_t^i = (\sigma_{x,t}^i, \sigma_{y,t}^i)$, and ρ_t^i is the correlation coefficient. In this work, we train our model to learn the estimated trajectory distribution (i.e., $\mu_{x,t}^i, \mu_{y,t}^i, \sigma_{x,t}^i, \sigma_{y,t}^i, \rho_t^i$), and then sample the estimated distribution to obtain the upcoming trajectories of pedestrians.

3.2. Dual-branch spatio-temporal graph network

As shown in Fig. 2, the proposed method consists of two branches, namely an STGAT branch used to model pedestrian view area (asymmetric social interactions between pedestrians) and an STGCN branch used to handle grouping (symmetric social interactions between pedestrians). Each branch generates an embedding of the input sequence, which encodes different social interactions. These two embeddings are then fused to form the final feature before being sent to the TCN module. The TCN module

performs trajectory prediction based on the fused feature using a convolution operation in the temporal dimension.

We consider pedestrians as nodes of graphs and the social interactions of pedestrians as edges between these nodes. We denote a graph as $G(V, E)$, where V denotes nodes and E denotes the edges between these nodes. In this work, we use two types of graphs to describe the same scene, i.e., directed graphs $G_{directed}$ and undirected graphs $G_{undirected}$. The corresponding adjacency matrices are $A_{directed}$ and $A_{undirected}$, respectively. The element a_{ij} in the adjacency matrix denotes an edge pointing from node i to node j . If the edge exists, then a_{ij} equals 1, otherwise, a_{ij} is 0. Note that $G_{directed}$ and $G_{undirected}$ represent a set of spatial graphs at different time steps, i.e., $G_{directed,t}$ and $G_{undirected,t}$. Correspondingly, $A_{directed}$ and $A_{undirected}$ are a set of $A_{directed,t}$ and $A_{undirected,t}$, respectively. The view area and group information are encoded in these adjacency matrices.

3.3. View area modeling in STGAT branch

The goal of the STGAT branch is to model pedestrian view areas and compute features for each pedestrian. This is achieved by aggregating features from neighbors (inside the view area) with different attention weights. In this way, the view area and different contributions of neighbors are considered, which can model social interactions more accurately. Moreover, spatial and temporal convolution are performed to aggregate spatial and temporal information, respectively. This branch is inspired by the STGAT proposed by Huang et al. [14]. Compared with STAGT, our method explicitly considers view area information and utilizes a TCN module to perform trajectory prediction.

3.3.1. Definition of view area

We describe the view area using the walking direction and view angle of a pedestrian², as shown in Fig. 3. There are three pedestrians in the scene of Fig. 3, $\mathbf{v}_1$ is the walking direction of pedestrian 1, $\mathbf{D}_{1i}$ is the vector pointing from pedestrian 1 to pedestrian i . As can be seen, when the angle between $\mathbf{v}_1$ and $\mathbf{D}_{1i}$ is smaller than a half of the view angle of pedestrian 1, the pedestrian i is in the

² We decide the view area based on walking direction rather than eye gaze direction. We use the name of 'view area' for convenience.

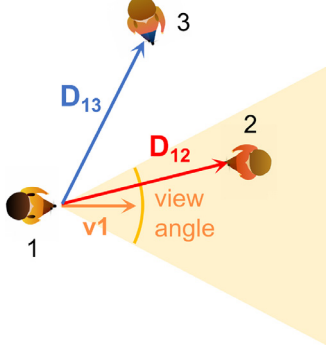


Fig. 3. Finding the view area of a pedestrian.

view area of pedestrian 1. Accordingly, in Fig. 3, pedestrian 2 is in the view area of pedestrian 1, while pedestrian 3 is not.

In this work, we design a “Finding View Area Module” in the STGAT branch to automatically find the view area of each pedestrian and the pedestrians inside each view area. Note that the positions of pedestrians may change continuously. To achieve better prediction, we should keep an eye on everyone inside the view area, although other pedestrians might occlude a pedestrian temporally. The pedestrians inside the view area will be encoded in the adjacency matrices of the STGAT to describe these social interactions.

It should be noted that the MX-LSTM proposed by Hasan et al. [20] and its extension [21] use head pose to define the view area. However, there are several differences between MX-LSTM and our method. First, MX-LSTM builds the view area based on the head pose, which is obtained from an additional head pose detector or manual annotation. However, the head pose is not always consistent with walking direction. For example, one pedestrian can look around while walking. Therefore, the head pose may not necessarily reflect the walking direction. By contrast, our method adopts the walking direction obtained from positions in sequences to automatically find the view area. Second, MX-LSTM uses a fixed value of 144° in the definition of view area. By contrast, in this work, we show that the view angle affects the performance and find a relatively better angle via experiments. Third, in MX-LSTM, the view area is first divided into different cells, and then a pooling is applied. In our method, a GAT is adopted to handle the pedestrians inside the view area to automatically set different weights to them. Fourth, we model group behaviors using STGCN, while MX-LSTM does not consider group behaviors. Finally, we consider input sequences as spatial-temporal graph data, while MX-LSTM uses the traditional LSTM network. Because of these differences, our method has better performance than MX-LSTM, as shown in Section 4.3.

3.3.2. Calculation of walking direction

We calculate the walking direction of pedestrians automatically using their positions in consecutive frames. Specifically, if the positions of a pedestrian in two consecutive frames are $\mathbf{P}_1$ and $\mathbf{P}_2$, then the walking direction (vector) is $(\mathbf{P}_2 - \mathbf{P}_1)$. This method is simple yet effective. In this way, we can obtain the walking direction automatically and do not need additional detectors.

3.3.3. Graph convolution in spatial dimension

After obtaining the information of the view area, directed graphs are created for the scenes at each time step. In terms of implementation, the view area is encoded in the adjacency matrix $\mathbf{A}_{\text{directed}}$ of the directed graphs. Specifically, if a node j is in the view area of node i at time t , then the element a_{ji} is set to 1 in the adjacency matrix $\mathbf{A}_{\text{directed},t}$. Otherwise, the element a_{ji} is set to

0. Consequently, $\mathbf{A}_{\text{directed}}$ is a set of asymmetric matrices. An example of the adjacency matrix at one time step is shown at the right corner of Fig. 2.

To encode view area information, we first run graph convolution in the spatial dimension. Specifically, GAT operates on $G_{\text{directed},t}$ and computes the feature of each node. This is achieved by aggregating information from its neighbors, including itself, using a self-attention strategy. The GAT network consists of graph attention layers. For the simplicity of notation, we omit the subscript t for most notations in this subsection. Assume the input of the graph attention layer l is $\mathbf{h}^l = \{\mathbf{h}_1^l, \mathbf{h}_2^l, \dots, \mathbf{h}_N^l\}$, where $\mathbf{h}_i^l \in \mathbb{R}^F$, F is the feature dimension of each node, N is the number of nodes. The output of the graph attention layer l , i.e., the input of layer $(l+1)$, is $\mathbf{h}^{(l+1)} = \{\mathbf{h}_1^{(l+1)}, \mathbf{h}_2^{(l+1)}, \dots, \mathbf{h}_N^{(l+1)}\}$, where $\mathbf{h}_i^{(l+1)} \in \mathbb{R}^{F'}$, F' and F can be different. The features used as input to the first layer are the 2D relative positions of pedestrians.

In this work, we argue that the motion of a pedestrian is affected by pedestrians in his/her view area and is usually not affected by pedestrians outside the view area. Therefore, the attention coefficients of node j to i in layer l can be computed as

$$\alpha_{i,j}^l = \frac{\exp(\text{LeakyReLU}(a^T [\mathbf{W}^l \mathbf{h}_i^l || \mathbf{W}^l \mathbf{h}_j^l]))}{\sum_{k \in V_i} \exp(\text{LeakyReLU}(a^T [\mathbf{W}^l \mathbf{h}_i^l || \mathbf{W}^l \mathbf{h}_k^l]))}, \quad (2)$$

where V_i is the view area of node i embedded in $\mathbf{A}_{\text{directed},t}$, a is weight vector, $||$ is concatenation operation, T denotes transposition, $\mathbf{W}^l \in \mathbb{R}^{F' \times F}$ is the weight matrix of a shared linear transformation that is applied to each node. The attention coefficients are normalized by a softmax function.

In terms of implementation, we first compute the attention coefficients between all nodes and then use $\mathbf{A}_{\text{directed},t}$ as a mask to obtain the results of Eq. (2). After computing the normalized attention coefficients, the output of node i of graph attention layer l is:

$$\mathbf{h}_i^{(l+1)} = \sigma \left(\sum_{j \in V_i} \alpha_{i,j}^l \mathbf{W}^l \mathbf{h}_j^l \right), \quad (3)$$

where σ is LeakyReLU [40] and the negative slope is 0.2. This is the output of single graph attention layer. In practice, several graph attention layers can be used to obtain $\mathbf{F}_{\text{GAT},t}$, i.e., the output feature for the graph at time step t , whose dimension is $5 \times N$. We model the trajectory using bivariate Gaussian distribution, so here ‘5’ means the five parameters of the bivariate Gaussian distribution (Eq. (1)), i.e., $\mu_{x,t}^i, \mu_{y,t}^i, \sigma_{x,t}^i, \sigma_{y,t}^i, \rho_t^i$. Note that in this work, multi-head attention is employed. Specifically, the number of heads is 4.

3.3.4. Convolution in temporal dimension

After performing convolution in spatial dimension to encode information of the same time step, a standard convolution in the temporal dimension is performed to exploit temporal dependencies from nearby times. Specifically, the input to the temporal convolution is a tensor with dimension is $T_{\text{obs}} \times 5 \times N$ because it receives features from T_{obs} frames. Moreover, the temporal convolution is designed to output $\mathbf{F}_{\text{STGAT}}$ with the dimension of $T_{\text{obs}} \times 5 \times N$, as shown in Fig. 2.

In summary, after applying STGAT, we have the embedding feature $\mathbf{F}_{\text{STGAT}}$ that represents the directed graph G_{directed} and view area interaction. This work implements the view area in the STGAT branch, where attention coefficients are assigned to each pedestrian inside the view area. Although we do not explicitly consider the distance factor, the model can learn to set different attention coefficients to each pedestrian inside the view area. This actually considers the distance factor implicitly. Moreover, because we encode view area information in each time step, the adjacency matrix at each time step $\mathbf{A}_{\text{directed},t}$ might be different.

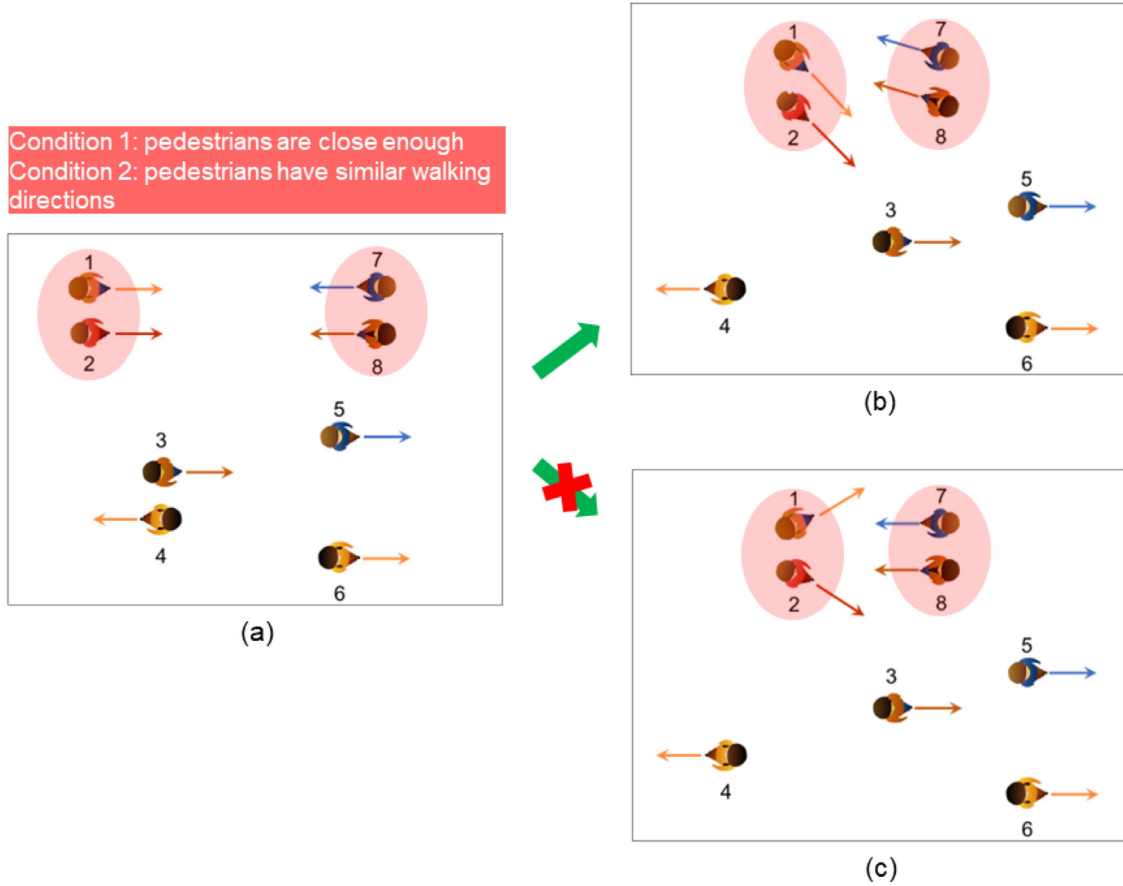


Fig. 4. The pedestrians in a group tend to have similar trajectories. (a) We use two conditions to judge whether two pedestrians are in a group. Specifically, pedestrians should be close enough and have similar walking directions. Accordingly, pedestrians 1 and 2 are in a group, and pedestrians 7 and 8 are also in a group. (b) Pedestrians 1 and 2 tend to have similar trajectories when avoiding collision with pedestrians 7 and 8. (c) It is less likely that pedestrian 1 and pedestrian 2 will walk separately.

3.4. Group modeling in STGCN branch

When walking in groups, pedestrians usually keep a short distance from each other and try to walk side by side [41]. Therefore, they tend to have similar motion patterns, as shown in Fig. 4. In this work, we first judge whether two pedestrians are in the same group using two conditions and then encode the group information in the STGCN via adjacency matrices. Furthermore, spatial and temporal convolutions are performed to aggregate spatial and temporal information, respectively. The design of our STGCN branch is based on Social STGCNN [15]. Compared with Social STGCNN, we explicitly encode group relations in the spatial-temporal graph.

3.4.1. Definition of groups

There are different ways to define groups. In this work, based on our logical analysis of walking behaviors, we consider pedestrians who are close and have similar walking directions as a group. First, the distance between these pedestrians should be short enough, such as pedestrians 7 and 8 in Fig. 1(e). Second, the walking directions of these pedestrians should be similar, i.e., the angle difference between the walking directions should be smaller than a threshold. In Fig. 1(e), although pedestrians 1 and 8 are close to each other, they are not in a group because they walk in opposite directions. It should be noted that we detect groups automatically and continuously from the second frame and update the adjacency matrices dynamically. Therefore, we do not require that a group is always present and we can handle group forming or changes. This definition enables us to find groups automatically in each frame and gives us good performance.

3.4.2. Finding groups

We design a “Finding Group Module” in the STGCN branch to automatically find groups as per our definition of groups. Specifically, we consider pedestrians i and j are in the same group at time t if they are close enough and have similar walking directions, i.e.,

$$\begin{cases} \mathbf{P}_{i,t} - \mathbf{P}_{j,t} < S_{\text{position}}, \\ \theta(\mathbf{V}_{i,t}, \mathbf{V}_{j,t}) < S_{\text{direction}}, \end{cases} \quad (4)$$

where $\mathbf{P}_{i,t}$ and $\mathbf{P}_{j,t}$ are the positions of the two pedestrians at time t , respectively. $\mathbf{V}_{i,t}$ and $\mathbf{V}_{j,t}$ are the walking directions of these two pedestrians computed according to the method proposed in Section 3.3.2. θ is angle difference. S_{position} and $S_{\text{direction}}$ are position threshold value and direction threshold value, respectively.

3.4.3. Graph convolution in spatial dimension

Group information is encoded in the adjacency matrix $\mathbf{A}_{\text{undirected}}$ of the undirected graph. $\mathbf{A}_{\text{undirected}}$ is obtained using information from the “Finding Group Module” of the STGCN branch. Specifically, if nodes i and j are in a group, then elements a_{ij} and a_{ji} are set to 1 in $\mathbf{A}_{\text{undirected},t}$. Otherwise, the elements are set to 0. Unlike $\mathbf{A}_{\text{directed},t}$, $\mathbf{A}_{\text{undirected},t}$ is symmetric. This is a very important difference between undirected graph and directed graph. An example of the adjacency matrix at one time step is shown in the right corner of Fig. 2.

For simplicity of notation, we omit the subscript t for most notations in this subsection. We denote the input of the GCN layer l as $\mathbf{H}^l$, then the output of layer l , i.e., the input of layer $(l+1)$, is

$$\mathbf{H}^{(l+1)} = \sigma'(\Lambda_t^{-\frac{1}{2}} \hat{\mathbf{A}}_{\text{undirected},t} \Lambda_t^{-\frac{1}{2}} \mathbf{H}^{(l)} \mathbf{W}^{l,l'}), \quad (5)$$

where $\hat{\mathbf{A}}_{\text{undirected},t} = \mathbf{A}_{\text{undirected},t} + \mathbf{I}$, Λ_t is the degree matrix³ of $\hat{\mathbf{A}}_{\text{undirected},t}$, $\mathbf{W}^{l'}$ is the matrix of trainable parameters of layer l' , σ' is PReLU [42] and the initial parameter is 0.25. The input of the first layer of GCN network is the same as the input of the first layer of GAT network, i.e., $\mathbf{H}^{(0)} = \mathbf{h}^{(0)}$. In practice, there can be several layers in the GCN. The output feature $\mathbf{F}_{\text{GCN},t}$ for the graph at time step t can be obtained by applying GCN to $G_{\text{undirected},t}$. The dimension of $\mathbf{F}_{\text{GCN},t}$ is $5 \times N$.

3.4.4. Convolution in temporal dimension

Similar to the STGAT branch, in the STGCN branch, spatial graph convolution is extended to spatio-temporal graph convolution. This is achieved by applying a standard convolution in the temporal dimension to utilize temporal information after performing the graph convolution in the spatial dimension. Specifically, the input to the temporal convolution is a tensor with dimension $T_{\text{obs}} \times 5 \times N$ because it receives features from T_{obs} frames. Moreover, the temporal convolution is designed to output $\mathbf{F}_{\text{STGCN}}$ with the dimension $T_{\text{obs}} \times 5 \times N$, as shown in Fig. 2. Here, '5' means the five parameters of the bivariate Gaussian distribution (Eq. (1)), i.e., $\mu_{x,t}^i, \mu_{y,t}^i, \sigma_{x,t}^i, \sigma_{y,t}^i, \rho_t^i$.

In summary, after applying STGCN, we have the embedding feature $\mathbf{F}_{\text{STGCN}}$ that encodes the undirected graph $G_{\text{undirected}}$ and group interaction. In this study, because we encode group information in each time step, the adjacency matrix at each time step $\mathbf{A}_{\text{undirected},t}$ might be different. More details about the spatio-temporal graph convolution can be found in [43].

3.5. Feature fusion

In the proposed method, the view area and groups are modeled dynamically and encoded in dynamic spatio-temporal graphs through adjacency matrices. Therefore, both embeddings $\mathbf{F}_{\text{STGAT}}$ and $\mathbf{F}_{\text{STGCN}}$ contain spatio-temporal information as well as social interactions. These two features are fused and sent to the temporal convolution module as

$$\mathbf{F}_{\text{fused}} = \mathbf{F}_{\text{STGAT}} \oplus \mathbf{F}_{\text{STGCN}}, \quad (6)$$

where $\oplus$ means the fusion operation. In this work, element-wise averaging is chosen. The dimension of $\mathbf{F}_{\text{fused}}$ is also $T_{\text{obs}} \times 5 \times N$.

3.6. Time convolution for trajectory prediction

Unlike most studies that utilize LSTM [10,19] to predict trajectories, we use a time convolution module (TCN) to perform prediction based on the fused feature $\mathbf{F}_{\text{fused}}$ following Social-STGCNN [15]. Specifically, the TCN is a series of residual connected CNNs. It operates on the temporal direction of $\mathbf{F}_{\text{fused}}$ and expands it for prediction. It receives features with dimension $T_{\text{obs}} \times P \times N$ and outputs a prediction with dimension $(T_{\text{pred}} - T_{\text{obs}}) \times P \times N$, where P is the feature dimension of pedestrian, N is the number of pedestrians. In this work, T_{pred} is 20, T_{obs} is 8, P is 5.

3.7. Loss function

Our model is trained to learn the bivariate Gaussian $\mathcal{N}(\mu_t^i, \sigma_t^i, \rho_t^i)$ of pedestrian trajectory. To this end, the parameters of the proposed model are learned by minimizing the negative log-likelihood loss, as in [15]. The loss is defined as:

$$L^i(\mathbf{W}) = - \sum_{t=T_{\text{obs}}+1}^{T_{\text{pred}}} \log(\mathbb{P}(x_t^i, y_t^i | \mu_t^i, \sigma_t^i, \rho_t^i)), \quad (7)$$

³ This is a diagonal matrix containing information about the degree of each node

where L^i is for the i th pedestrian and $\mathbf{W}$ includes all trainable parameters. We train the proposed model by minimizing this loss for all the trajectories in the training dataset.

4. Experiments

4.1. Datasets and metrics

Datasets: Following previous studies [10,11,14,15,27], we evaluated our method on two public datasets, i.e., ETH [41] and UCY [44]. The ETH dataset includes two scenes (ETH and HOTEL), and the UCY dataset consists of three scenes (UNIV, ZARA1, and ZARA2). These sequences are recorded in 25 frames per second (FPS) with fixed cameras. The scene backgrounds contained in these datasets are shown in Fig. 5. The average pedestrian number of a scene is 5.9 for ETH and 18.0 for UCY. All trajectory coordinates are converted to world coordinates and interpolated to sample the coordinate every 0.4 seconds [10]. These datasets consist of rich human-human interactions in the real world, such as collision avoidance and groups. In total, there are 1536 pedestrians covering complex social interactions. More details about these datasets can be found in the corresponding papers [41,44].

Following previous studies, we use the leave-one-out strategy for evaluation, i.e., we train the model on four sets and test on the remaining one. In addition, we take 8 frames (3.2 seconds) as observation and predict the next 12 time steps (4.8 seconds). These settings have been adopted by most studies in the field of pedestrian trajectory prediction.

Metrics: Following prior studies [10,11,14,15,27], we evaluate our method with two error metrics in meters.

- Average Displacement Error (ADE): Averaged Euclidean distance between ground truth and predictions over all predicted time steps for N pedestrians. It is defined as:

$$\text{ADE} = \frac{\sum_{i \in N} \sum_{t \in (T_{\text{pred}} - T_{\text{obs}})} \|(x_t^i, y_t^i) - (\hat{x}_t^i, \hat{y}_t^i)\|_2}{N \times (T_{\text{pred}} - T_{\text{obs}})} \quad (8)$$

- Final Displacement Error (FDE): Euclidean distance between ground truth and predictions at the last frame. It is defined as:

$$\text{FDE} = \frac{\sum_{i \in N} \|(x_{T_{\text{pred}}}^i, y_{T_{\text{pred}}}^i) - (\hat{x}_{T_{\text{pred}}}^i, \hat{y}_{T_{\text{pred}}}^i)\|_2}{N} \quad (9)$$

Because the proposed Social TAG generates a bivariate Gaussian distribution as the prediction, we employ the method adopted by many pedestrian trajectory prediction algorithms, such as Social LSTM [10] and Social-STGCNN [15], to compare a distribution with a certain target value. Specifically, 20 samples are generated from the estimated distribution and the closest sample to the ground truth is used to calculate ADE and FDE.

4.2. Implementation details

In this work, one STGAT layer is used in the STGAT branch, and one STGCN layer is used in the STGCN branch. In addition, five layers are used in the TCN module. The view angle is set to 25° in the STGAT branch. In the STGCN branch, the position threshold S_{position} and direction threshold $S_{\text{direction}}$ are set to 2 m and 3°, respectively. The batch size is 128. The learning rate is 0.01 for the first 150 epochs and is 0.001 for the rest 100 epochs. The model is trained for 250 epochs using Stochastic Gradient Descent (SGD). We train the model in the training set and select the model using the validation set. All the training, validation, and test sets are the same as previous studies in this field [10,11,15]. All training

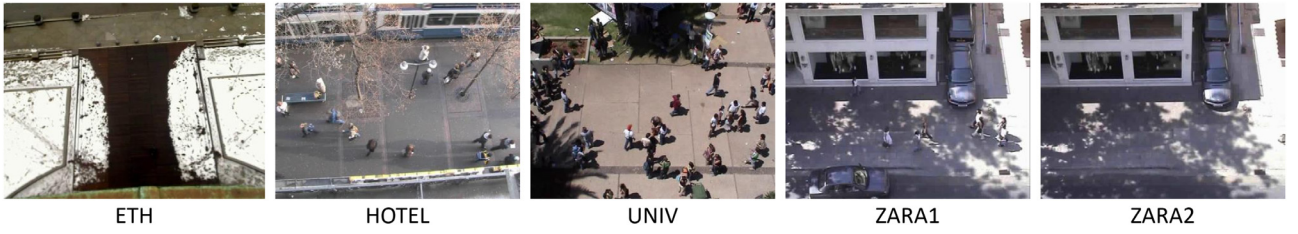


Fig. 5. Example frames from the ETH [41] and UCY datasets [44].

Table 1

ADE/FDE metrics (in meters) on ETH and UCY datasets. The best results in each column are marked in boldface and the second best results are underlined. The results marked with † are extracted from the corresponding papers and the results marked with § are copied from [11].

Method	Venue	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVERAGE
Social LSTM [10]§	CVPR2016	1.09/2.35	0.79/1.76	0.67/1.40	0.47/1.00	0.56/1.17	0.72/1.54
Social GAN-P [11]†	CVPR2018	0.87/1.62	0.67/1.37	0.76/1.52	0.35/0.68	0.42/0.84	0.61/1.21
Social-BiGAT [13]†	NeurIPS2019	0.69/1.29	0.49/1.01	0.55/1.32	<u>0.30/0.62</u>	0.36/0.75	0.48/1.00
SAGCN [32]†	ITNEC2019	0.90/1.96	0.41/0.83	0.57/1.19	0.41/0.89	0.32/0.70	0.52/1.11
STGAT [14]†	ICCV2019	0.65/1.12	0.35/0.66	0.52/1.10	0.34/0.69	<u>0.29/0.60</u>	0.43/0.83
SoPhie [45]†	CVPR2019	0.70/1.43	0.76/1.67	0.54/1.24	<u>0.30/0.63</u>	0.38/0.78	0.54/1.15
SR-LSTM [27]†	CVPR2019	0.63/1.25	0.37/0.74	0.51/1.10	0.41/0.90	0.32/0.70	0.45/0.94
NEXT [26]†	CVPR2019	0.73/1.65	<u>0.30/0.59</u>	0.60/1.27	0.38/0.81	0.31/0.68	0.46/1.00
PMP-NMMP [16]†	CVPR2020	<u>0.61/1.08</u>	0.33/0.63	0.53/1.17	0.28/0.61	0.28/0.59	0.41/0.82
Sun et al. [46]†	CVPR2020	0.69/1.24	0.43/0.87	0.52/1.11	0.32/0.66	<u>0.29/0.61</u>	0.44/0.90
RSBG [19]†	CVPR2020	0.80/1.53	0.33/0.64	0.59/1.25	0.40/0.86	0.30/0.65	0.48/0.99
Social-STGCNN [15]†	CVPR2020	0.64/1.11	0.49/0.85	0.44/0.79	0.34/ <u>0.53</u>	0.30/ 0.48	<u>0.44/0.75</u>
CoMoGCN [18]†	BMVC2020	0.70/1.26	0.37/0.75	0.53/1.16	0.34/0.71	0.31/0.67	0.45/0.91
DSCMP [47]†	ECCV2020	0.71/1.35	0.33/0.58	0.60/1.28	0.38/0.81	0.49/1.09	0.50/1.01
FvTraj [48]†	ECCV2020	0.56/1.14	0.28/0.55	0.52/1.12	0.37/0.78	0.32/0.68	0.41/0.85
SGC_LSTM [31]†	IET ITS2021	0.82/1.72	0.32/0.62	<u>0.50/1.10</u>	0.40/0.87	0.32/0.71	0.47/1.01
RAMP [49]†	T-CSVT2021	0.69/1.24	0.43/0.87	0.53/1.17	0.28/0.61	0.28/0.59	0.44/0.90
Social TAG (ours)		<u>0.61/1.00</u>	0.37/ <u>0.56</u>	0.51/ <u>0.87</u>	0.33/ 0.50	0.30/ <u>0.49</u>	<u>0.42/0.68</u>

and experiments are performed using an NVIDIA RTX 3090 GPU and an i9-10900X CPU.

4.3. Comparison with existing methods

We compare the proposed Social TAG with 17 existing methods, including: Social LSTM [10], Social GAN-P [11], Social-BiGAT [13], SoPhie [45], SR-LSTM [27], SAGCN [32], STGAT [14], NEXT [26], PMP-NMMP [16], Sun et al. [46], RSBG [19], Social-STGCNN [15], CoMoGCN [18], DSCMP [47], FvTraj [48], SGC_LSTM [31], RAMP [49]. The comparison results of ADE and FDE are shown in Table 1. As can be seen, the proposed Social TAG obtains very competitive performance compared with the state-of-the-art methods. Specifically, Social TAG obtains the best average FDE value (with 7% improvement) among all methods and achieves the second best in terms of ADE. From the table, we can see that the value of ADE (0.42m) is smaller than FDE (0.68m), indicating that our method can well predict all time steps. Also, the better ranking of FDE (1st) than ADE (2nd) indicates that our method can better predict the final step compared with other methods. In addition, the proposed Social TAG achieves the best value in two metric values and is ranked the second place in four metric values on five sets. The competitive performance of Social TAG is due to the view area-based STGAT design and group-based STGCN design, which not only model both asymmetric and symmetric interactions, but also reduce noise information.

4.4. Ablation study

4.4.1. Component analysis

In this section, we investigate the effect of different components proposed in this work. Table 2 lists the prediction errors of

Social TAG with different configurations. The Baseline means that there is no view area applied in the STGAT branch and no group behavior modeled in the STGCN branch. The Finding View Area module is removed in Configuration 2, while the Finding Group module is removed in Configuration 3. All components are used in Configuration 4. Table 2 indicates that both the Finding View Area module and the Finding Group module contribute positively to the best overall performance of the proposed method. The reason is that the modeling of view area reduces noise in features and the modeling of groups captures similar motion patterns for pedestrians walking in a group that is very frequent in the pedestrian trajectory prediction task.

4.4.2. Effect of view angle on prediction errors

The view angle affects the number of pedestrians inside the view area of a pedestrian, thus affecting the element values in $\mathbf{A}_{\text{directed}}$. To investigate the effect of view angle on prediction errors, we conducted a series of experiments based on the proposed Social TAG with different view angles. Considering the possible head movement, we also included a view angle of 180° . The threshold values S_{position} and $S_{\text{direction}}$ used to find groups are 2 m and 3° , respectively. As shown in Table 3, the view angle indeed affects the trajectory prediction performance. From the table one can see that a view angle of 25° gives the best overall performance among all investigated view angles. Also, the errors generally become larger as the view angle increases. This is because when a pedestrian is walking, only the pedestrians ahead of him/her and within a small view angle will affect his/her trajectory. Other pedestrians that may be seen by him/her but not inside the small view angle will not affect the trajectory, as shown in Fig. 6. Indeed, when the view angle is too large, pedestrians that should not influence a person are also taken into account when learning

Table 2

Results (in meters) with different configurations. Configuration 2 does not have the Finding View Area Module, and Configuration 3 does not have the Finding Group module. Baseline does not have both modules, and Configuration 4 has both modules. The view angle is 25°. Grouping parameters are 2 m and 3°, respectively. The best results are marked in boldface.

Version	Components		Results (ADE/FDE)					
	Finding Group	Finding View Area	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVERAGE
Baseline	✗	✗	0.631/1.162	0.478/0.802	0.509/0.915	0.379/0.636	0.348/0.569	0.469/0.817
Configuration 2	✓	✗	0.652/1.073	0.433/0.693	0.529/0.929	0.369/0.579	0.335/0.554	0.464/0.766
Configuration 3	✗	✓	0.626/1.035	0.415/0.659	0.493/0.867	0.358/0.556	0.318/0.519	0.442/0.727
Configuration 4	✓	✓	0.605/0.996	0.367/0.556	0.507/0.871	0.333/0.504	0.300/0.490	0.422/0.683

Table 3

Effect of view angle in the STGAT branch on prediction errors (in meters) on ETH and UCY datasets. Grouping parameters are 2 m and 3°, respectively. The best results are marked in boldface.

View angle	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	90°	120°	150°	180°
ADE	0.441	0.432	0.422	0.428	0.430	0.431	0.430	0.430	0.438	0.440	0.447	0.447	0.455	0.455
FDE	0.737	0.710	0.683	0.698	0.701	0.701	0.692	0.705	0.710	0.701	0.724	0.725	0.740	0.765

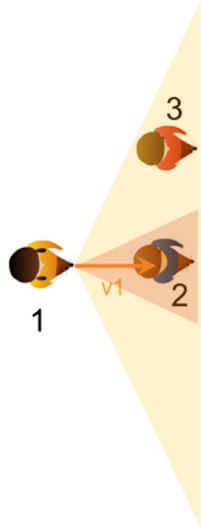


Fig. 6. Although pedestrian 3 is within the view area of pedestrian 1, he or she may not affect the trajectory of pedestrian 1. In contrast, pedestrian 2, who is directly ahead of pedestrian 1, is considered by pedestrian 1.

the embedding of the scene, thus introducing noise and leading to performance degradation. Furthermore, the view angle should not be too small. As shown in Table 3, when the view angle is 15°, the results became much worse than those of 25°. This is because that when the view angle is too small, the pedestrians in the view area will be too few, thus it is unable to model social interactions properly.

To further investigate the effect of view angle, we use different grouping parameters and see how the view angle will affect the prediction results. The results are shown in Fig. 7. As can be seen, in most configurations of grouping parameters, the prediction error will generally be larger if the view angle is too large.

4.4.3. Effect of grouping conditions

The parameters used to find groups (see Eq. (4)), i.e., position threshold and direction threshold, affect the grouping results and therefore result in different prediction performance. To investigate the effect of these parameters on prediction errors, we performed a grid search study and put the results in Table 4. As can be seen, these two parameters have an impact on the prediction errors. Generally speaking, these two parameters can neither be

too large nor too small. If they are too large, more noise will be contained when extracting pedestrian features through the STGCN branch. By contrast, if they are too small, the conditions will be too strict to properly model group interactions between pedestrians. Among all combinations of parameters investigated in this work, when the view angle is 25°, the position threshold is 2 m and the direction threshold is 3°, the best overall performance is obtained.

It is worth mentioning that the ablation studies on view angle and grouping parameters aim to show the general effect of these parameters. These values might vary slightly in practice if the number and distribution of pedestrians are different.

4.4.4. Effect of feature fusion method

Effective fusion of STGAT and STGCN features is crucial for the proposed method. To investigate the effect of feature fusion, we performed a series of experiments with different fusion methods. Specifically, we designed four variants. In the first variant (V1), we used element-wise multiplication between STGAT and STGCN features. In the second variant (V2), we used an STGAT-gated fusion method, that is,

$$\mathbf{F}_{\text{fused}} = \text{sigmoid}(\mathbf{F}_{\text{STGAT}}) \otimes \mathbf{F}_{\text{STGCN}}, \quad (10)$$

where $\otimes$ is element-wise multiplication. In this variant, the Sigmoid function was applied to every element of the STGAT feature to non-linearly scale it to the range (0,1). This has the effect as a gate on the encoded STGCN feature to control the passing or rejection of the STGCN feature.

In the third variant (V3), we utilized an STGCN-gated fusion method, i.e.,

$$\mathbf{F}_{\text{fused}} = \mathbf{F}_{\text{STGAT}} \otimes \text{sigmoid}(\mathbf{F}_{\text{STGCN}}), \quad (11)$$

where the STGCN feature with Sigmoid function applied has the effect of a gate. In the fourth variant (V4), we employed element-wise max pooling to fuse features, i.e.,

$$\mathbf{F}_{\text{fused}} = \text{element_max}(\mathbf{F}_{\text{STGAT}}, \mathbf{F}_{\text{STGCN}}), \quad (12)$$

where $\text{element_max}()$ chooses the max value in an element-wise manner.

The performance of different variants is listed in Table 5. As can be seen, the element-wise averaging fusion method, i.e., the one utilized in the proposed method, provides the best overall performance. Specifically, the proposed method outperforms V1, V2, and V4 by a large margin, indicating that these methods are not able to fuse the STGAT and STGCN features effectively. V3 has the

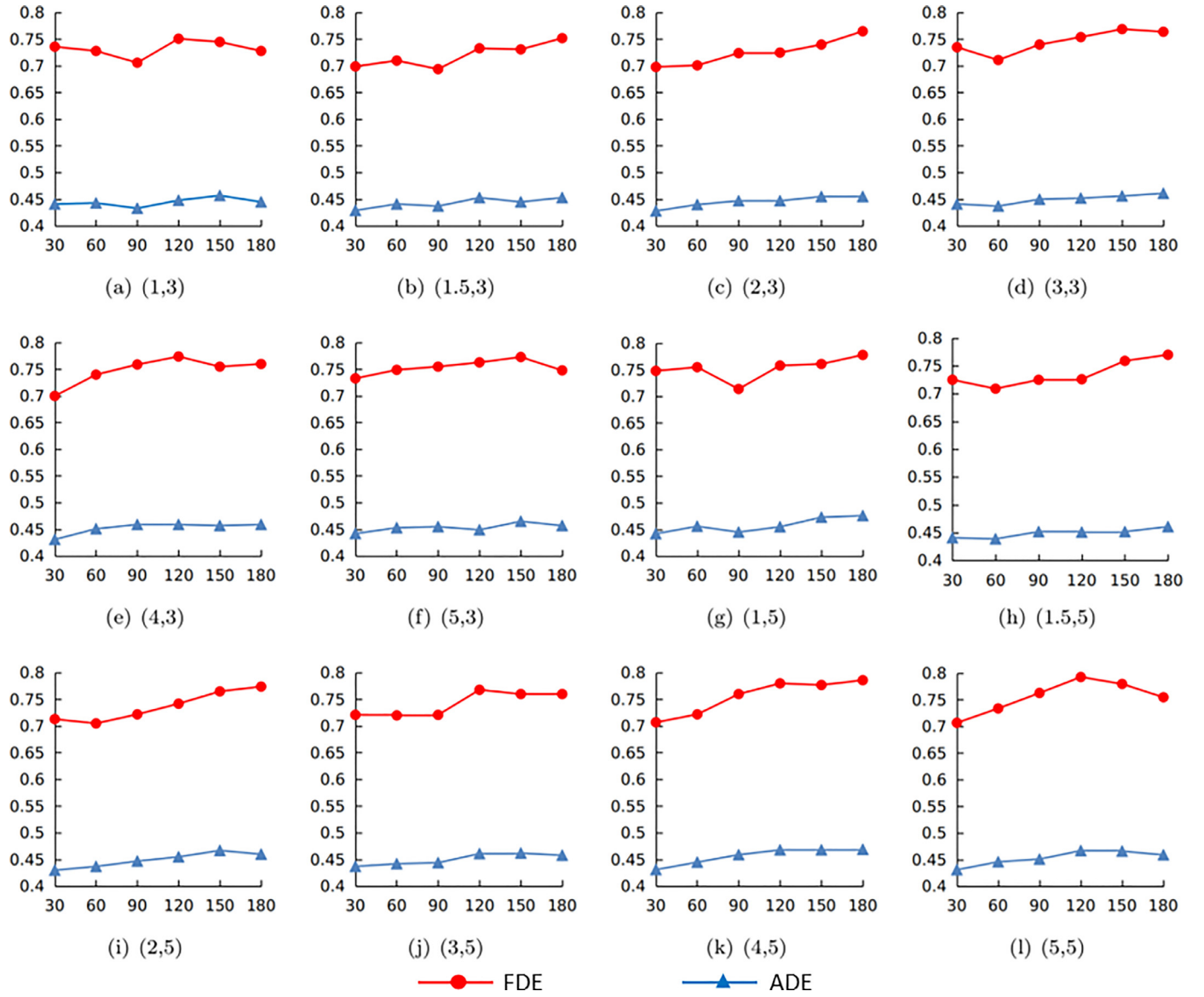


Fig. 7. The effect of view angle on prediction errors. The first value in the caption of each subfigure denotes the position threshold S_{position} and the second value denotes the direction threshold $S_{\text{direction}}$ in Equation (4). The horizontal axis is view angle and the vertical axis denotes error value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4

Effect of grouping parameters on prediction errors of Social TAG on ETH and UCY datasets. Average ADE/FDE metrics are shown in meters. The first row corresponds to the direction threshold between walking directions in degrees. The first column corresponds to the position threshold in meters. The best results in each column are marked in boldface and the second best results are underlined.

Direction threshold (°)	1	3	5	7	9	View angle
Position threshold (m)						
2	0.436/0.720	0.422/0.683	<u>0.428</u> /0.697	0.441/0.725	0.436/0.717	25°
1	0.431/0.718	0.441/0.736	0.442/0.748	0.440/0.738	0.440/0.740	30°
1.5	0.438/0.716	0.429/0.699	0.441/0.725	0.433/0.705	0.441/0.733	
2	0.434/0.712	<u>0.428</u> /0.698	0.430/0.713	0.441/0.731	0.434/0.709	
3	0.446/0.741	0.441/0.735	0.437/0.721	0.435/0.717	0.444/0.731	
4	0.449/0.750	0.431/0.700	0.431/0.707	0.431/0.704	0.437/0.711	
5	0.441/0.732	0.442/0.733	0.431/0.707	0.443/0.733	0.441/0.724	60°
1	0.440/0.703	0.443/0.728	0.456/0.755	0.434/0.720	0.436/0.700	
1.5	0.448/0.715	0.441/0.710	0.439/0.709	0.437/0.709	<u>0.436</u> /0.696	
2	0.441/0.708	0.440/0.701	0.437/0.705	0.442/0.709	0.437/0.698	
3	0.439/0.714	0.437/0.711	0.442/0.720	0.446/0.729	0.450/0.738	
4	0.451/0.749	0.451/0.740	0.445/0.722	0.449/0.732	0.445/0.722	
5	0.451/0.748	0.453/0.749	0.446/0.734	0.451/0.744	0.448/0.724	

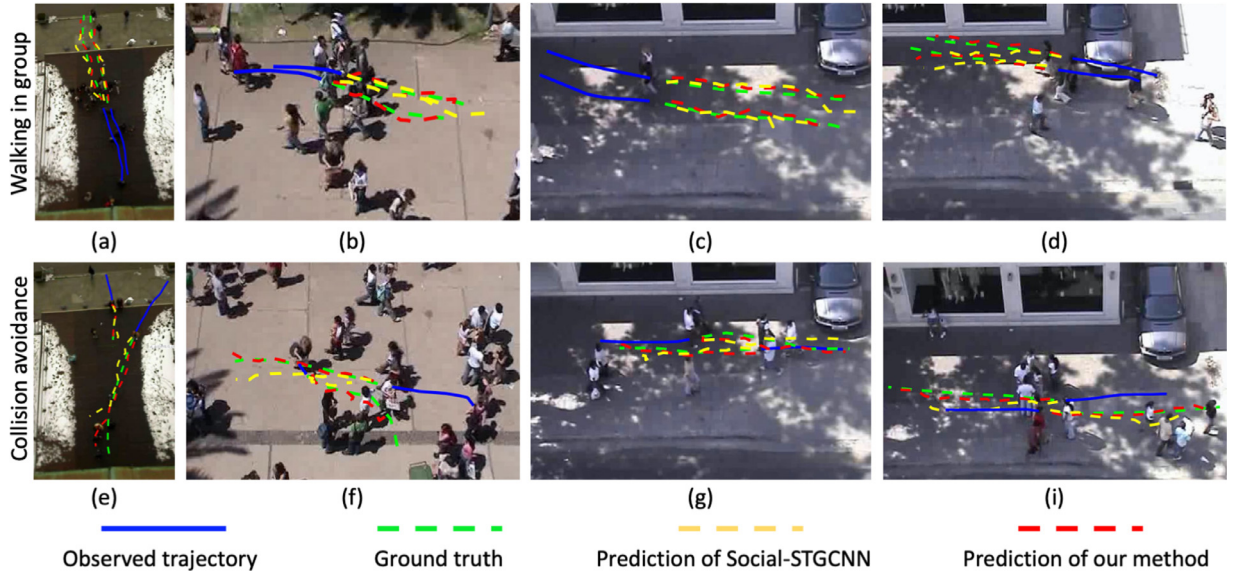


Fig. 8. Comparison of our method with Social-STGCNN in different scenarios. In the first row, we show examples where our model successfully predicts the trajectories of pedestrians walking in a group. In the second row, we show examples where our model successfully predicts the trajectories that avoid collision. Images are chosen from the ETH [41] and UCY [44], and results are generated in this study using the proposed Social TAG. Images were cropped for better visualization.

Table 5

Effect of feature fusion method on prediction errors (ADE/FDE metrics in meters) on ETH and UCY datasets. View angle is 25° , grouping threshold values are 2 m and 3° , respectively.

Variant	ADE	FDE
V1 (element-wise multiplication)	0.612	1.077
V2 (STGAT-gated fusion)	0.462	0.788
V3 (STGCN-gated fusion)	0.448	0.733
V4 (element-wise max pooling)	0.484	0.797
Social TAG (element-wise averaging)	0.422	0.683

closest performance to the proposed Social TAG. However, there is still a clear gap between its performance and that of the proposed method.

4.5. Qualitative results

In this section, we show some qualitative results (visualizations) to demonstrate the effectiveness of the proposed method.

4.5.1. Socially acceptable trajectory generation

Figure 8 shows the comparison of qualitative results between the proposed Social TAG and the state-of-the-art graph-based method Social-STGCNN in different scenarios. The first row indicates scenarios in which some pedestrians are walking in a group. As can be seen, our method can understand that these pedestrians are walking in a group and predict similar trajectories for them. By contrast, the Social-STGCNN cannot predict similar trajectories for these pedestrians. The second row illustrates the cases where pedestrians should avoid collision. From the figures, one can see that the proposed Social TAG can successfully avoid collision in these scenarios and predict trajectories similar to the ground truth. For examples, in cases (f) and (g), one pedestrian slightly changed his/her walking direction to avoid a collision with other pedestrians. By contrast, the trajectories predicted by Social-STGCNN deviate from the ground truth and do not capture the pattern of direction change due to collision avoidance. In summary, the proposed Social TAG can well model social interactions

between pedestrians, especially the view area (for collision avoidance) and the groups, thus can predict socially acceptable trajectories in different scenarios.

4.5.2. Other challenging cases

Figure 9 shows the qualitative comparison of our method with Social-STGCNN in several other challenging cases, including sudden movement, sudden stop, changing direction to avoid a collision, and nonlinear trajectory. As can be seen, our method gives superior performance than Social-STGCNN in these scenarios. The predicted trajectories are also very close to the ground truth trajectories. This is due to the effectiveness of the proposed spatio-temporal GNN-based methods with view area and grouping modeling.

5. Discussions

5.1. About social interactions

There are different ways to model social interactions in pedestrian trajectory prediction. Indeed, because social interactions are very complex, different researchers may have different understandings and considerations on social interactions. Even for the same concept (for example, group), different researchers have proposed various definitions [18,31,46]. However, no matter how social interactions are defined and modeled, the ultimate goal is to successfully predict pedestrian trajectories, which can be evaluated using standard benchmarks and metrics.

In the authors' opinion, when pedestrians are walking, the main factors they usually consider are who is ahead of them and whom they are walking with. These two factors can be modeled by view area (asymmetric interactions) and grouping (symmetric interactions), respectively. Therefore, in this study, we combine the view area and grouping simultaneously and automatically, which have not been combined in existing studies yet. Our experiments have shown the competitive performance of the proposed method, demonstrating the effectiveness of our consideration of social interactions.

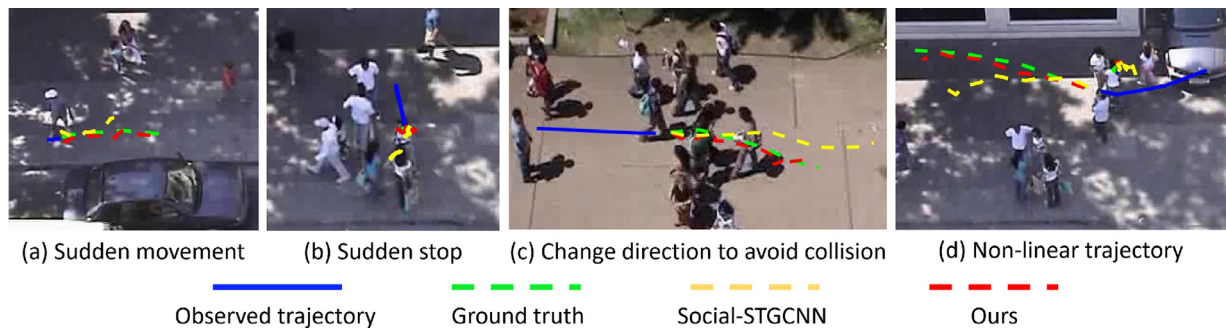


Fig. 9. Qualitative comparison of our method and Social-STGCNN in several challenging cases. (a) The pedestrian starts to move suddenly. (b) The pedestrian stops her motion suddenly. (c) The pedestrian changes his/her walking direction to avoid collision with other pedestrians. (d) The pedestrian has a highly non-linear trajectory. Our method successfully predicts motion patterns and trajectories for these challenging cases. The predicted trajectories are close to ground truth. Images are chosen from UCY [44], and results are generated in this study using the proposed Social TAG. Images were cropped for better visualization.

5.2. About hyperparameters

In the training of the proposed method, we first selected hyperparameters based on the recommendation of Mohamed et al. [15] and then refined the hyperparameters based on our training results. In this way, we finalized the hyperparameters stated in Section 4.2. Note that we did not perform an extensive hyperparameter tuning process because this is not the focus of this study.

The view angle and grouping parameters are also hyperparameters in the proposed method. We found the relatively better view angle and grouping parameters through a series of experiments on the chosen datasets, as shown in Sections 4.4.2 and 4.4.3. For different datasets, these values may vary slightly because the number and distribution of pedestrians may be different. In our future work, we will extend our method to select these values adaptively.

5.3. Ethical implications

Although person tracking in general can have important privacy implications, our specific method simply predicts the trajectory of pedestrians for the next 4.8 seconds without identifying them. To the best of our knowledge, we cannot think of any potential ethical aspects for which such short term anonymous trajectory prediction would need special consideration.

6. Conclusions

This paper proposed Social TAG, a novel dual-branch spatio-temporal graph neural network-based pedestrian trajectory method. The proposed Social TAG consists of two branches that model pedestrian view area and group behaviors, respectively. To further improve performance, a realistic and automatic Finding View Area module and a Finding Group module are designed in the corresponding branch. In this way, our Social TAG is able to model view area and group information simultaneously and automatically. A temporal convolutional network is then applied to predict trajectories. To the best of our knowledge, this is the first graph-based method that automatically models both view area and grouping. Evaluations on standard datasets show that our Social TAG achieves very competitive performance. Moreover, our visualizations show that the proposed method is able to capture interaction patterns and predict precise trajectories in different challenging scenarios.

Incorporating complex social interactions is challenging yet crucial in the pedestrian trajectory prediction task. This study introduces a new way of incorporating different types of social interactions into one model using spatio-temporal graph neural networks

and a multi-branch architecture, which can be helpful for other researchers in this field. Moreover, the explicit modeling of view area and grouping is also helpful for finding better trajectory patterns.

Although our method achieves promising performance, several things are worth investigating in our future work to further improve performance. First, we focus on social interactions between pedestrians in this study. However, context information (e.g., non-human obstacles) is also important in the pedestrian trajectory prediction task. In our future work, we will try to add context information to our method. A potential way is to add another branch to our framework to process context information and then fuse the context features with the STGAT and STGCN features. Second, we simply adopted element-wise averaging to fuse the STGAT and STGCN features. Although this shows competitive performance, we think it might further improve the performance by employing advanced feature fusion techniques. For example, the fusion operation can be made as a part of the learning framework and obtained through learning. Third, in this study, we are using existing datasets and assuming that each of the N pedestrians is identified and tracked throughout the input sequence. However, in practice, it is possible that the positions of pedestrians in some frames are missing. Therefore, it is worth developing prediction methods that are robust to information missing. Fourth, we assume that the number of pedestrians is fixed in the T_{obs} time steps following existing studies, thus the number of nodes of the STGCN and STGAT in the T_{obs} time steps is fixed as well. However, pedestrians may leave or enter the scene during the T_{obs} time steps in practice, thus it is worth investigating how to handle the varying number of nodes in spatio-temporal graphs. In addition, occlusion is an important factor that needs to be handled. For example, a pedestrian may be occluded by other pedestrians or obstacles and reappears. To solve this, we will need to combine multi-object tracking, pedestrian identification and pedestrian trajectory prediction techniques. Moreover, we manually select parameters for view angle and grounding conditions in this work. In the future, we will explore ways to set these parameters adaptively. Finally, we plan to apply the proposed method to trajectory prediction from egocentric view.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This work was supported in part by the InnovateUK DRISK project, in part by a Royal Academy of Engineering Chair in Emerging Technologies, and in part by the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie Grant agreement No 101025274.

References

- [1] F. Marchetti, F. Becattini, L. Seidenari, A. Del Bimbo, Multiple trajectory prediction of moving agents with memory augmented networks, *IEEE Trans. Pattern Anal. Mach. Intell.* (2020).
- [2] R. Quan, L. Zhu, Y. Wu, Y. Yang, Holistic LSTM for pedestrian trajectory prediction, *IEEE Trans. Image Process.* 30 (2021) 3229–3239.
- [3] Y. Cai, L. Dai, H. Wang, L. Chen, Y. Li, M.A. Sotelo, Z. Li, Pedestrian motion trajectory prediction in intelligent driving from far shot first-person perspective video, *IEEE Trans. Intell. Transp. Syst.* (2021).
- [4] A.J. Sathyamoorthy, J. Liang, U. Patel, T. Guan, R. Chandra, D. Manocha, DenseCAvoid: real-time navigation in dense crowds using anticipatory behaviors, in: *Proceedings of IEEE International Conference on Robotics and Automation*, IEEE, 2020, pp. 11345–11352.
- [5] Z. Chen, C. Song, Y. Yang, B. Zhao, Y. Hu, S. Liu, J. Zhang, Robot navigation based on human trajectory prediction and multiple travel modes, *Appl. Sci.* 8 (11) (2018) 2205.
- [6] C. Barata, J.C. Nascimento, J.A.M. Lemos, J.S. Marques, Sparse motion fields for trajectory prediction, *Pattern Recognit.* 110 (2021) 107631.
- [7] L. Rossi, M. Paolanti, R. Pierdicca, E. Frontoni, Human trajectory prediction and generation using LSTM models and GANs, *Pattern Recognit.* (2021) 108136.
- [8] Z. Pei, X. Qi, Y. Zhang, M. Ma, Y.-H. Yang, Human trajectory prediction in crowded scene using social-affinity long short-term memory, *Pattern Recognit.* 93 (2019) 273–282.
- [9] D. Helbing, P. Molnar, Social force model for pedestrian dynamics, *Phys. Rev. E* 51 (5) (1995) 4282.
- [10] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, S. Savarese, Social LSTM: human trajectory prediction in crowded spaces, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 961–971.
- [11] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, A. Alahi, Social GAN: socially acceptable trajectories with generative adversarial networks, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2255–2264.
- [12] S. Zamboni, Z.T. Kefato, S. Girdzijauskas, C. Norén, L. Dal Col, Pedestrian trajectory prediction with convolutional neural networks, *Pattern Recognit.* 121 (2022) 108252.
- [13] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. Reid, H. Rezatofighi, S. Savarese, Social-BiGAT: multimodal trajectory forecasting using bicycle-GAN and graph attention networks, in: *Proceedings of Advances in Neural Information Processing Systems*, 2019, pp. 137–146.
- [14] Y. Huang, H. Bi, Z. Li, T. Mao, Z. Wang, STGAT: modeling spatial-temporal interactions for human trajectory prediction, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6272–6281.
- [15] A. Mohamed, K. Qian, M. Elhoseiny, C. Claudel, Social-STGCNN: a social spatio-temporal graph convolutional neural network for human trajectory prediction, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14424–14432.
- [16] Y. Hu, S. Chen, Y. Zhang, X. Gu, Collaborative motion prediction via neural motion message passing, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6319–6328.
- [17] A. Vemula, K. Mueller, J. Oh, Social attention: modeling attention in human crowds, in: *Proceedings of IEEE international Conference on Robotics and Automation*, IEEE, 2018, pp. 1–7.
- [18] Y. Chen, C. Liu, B. Shi, M. Liu, CoMoGCN: coherent motion aware trajectory prediction with graph representation, in: *Proceedings of the British Machine Vision Conference*, 2020.
- [19] J. Sun, Q. Jiang, C. Lu, Recursive social behavior graph for trajectory prediction, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 660–669.
- [20] I. Hasan, F. Setti, T. Tsesmelis, A. Del Bue, F. Galasso, M. Cristani, MX-LSTM: mixing tracklets and vislets to jointly forecast trajectories and head poses, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6067–6076.
- [21] I. Hasan, F. Setti, T. Tsesmelis, V. Belagiannis, S. Amin, A. Del Bue, M. Cristani, F. Galasso, Forecasting people trajectories and head poses by jointly reasoning on tracklets and vislets, *IEEE Trans. Pattern Anal. Mach. Intell.* 43 (4) (2021) 1267–1278.
- [22] Z. Huang, J. Wang, L. Pi, X. Song, L. Yang, LSTM based trajectory prediction model for cyclist utilizing multiple interactions with environment, *Pattern Recognit.* 112 (2021) 107800.
- [23] A. Syed, B.T. Morris, Sseg-LSTM: semantic scene segmentation for trajectory prediction, in: *Proceedings of IEEE Intelligent Vehicles Symposium*, IEEE, 2019, pp. 2504–2509.
- [24] A.D. Berenguer, M. Alioscha-Perez, M.C. Oveneke, H. Sahli, Context-aware human trajectories prediction via latent variational model, *IEEE Trans. Circuits Syst. Video Technol.* 31 (5) (2021) 1876–1889.
- [25] J. Li, H. Ma, M. Tomizuka, Conditional generative neural system for probabilistic trajectory prediction, in: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2019, pp. 6150–6156.
- [26] J. Liang, L. Jiang, J.C. Niebles, A.G. Hauptmann, L. Fei-Fei, Peeking into the future: predicting future person activities and locations in videos, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5725–5734.
- [27] P. Zhang, W. Ouyang, P. Zhang, J. Xue, N. Zheng, SR-LSTM: state refinement for LSTM towards pedestrian trajectory prediction, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12085–12094.
- [28] F. Giuliani, I. Hasan, M. Cristani, F. Galasso, Transformer networks for trajectory forecasting, in: *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 10335–10342.
- [29] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, A. Gaidon, It is not the journey but the destination: endpoint conditioned trajectory prediction, in: *European Conference on Computer Vision*, Springer, 2020, pp. 759–776.
- [30] L. Zhang, Q. She, P. Guo, Stochastic trajectory prediction with social graph network, *arXiv preprint arXiv:1907.10233* (2019).
- [31] Y. Zhou, H. Wu, H. Cheng, K. Qi, K. Hu, C. Kang, J. Zheng, Social graph convolutional LSTM for pedestrian trajectory prediction, *IET Intel. Transport Syst.* 15 (3) (2021) 396–405.
- [32] Y. Sun, T. He, J. Hu, H. Huang, B. Chen, Socially-aware graph convolutional network for human trajectory prediction, in: *Proceedings of IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference*, IEEE, 2019, pp. 325–333.
- [33] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, *IEEE Trans. Neural Netw.* 20 (1) (2008) 61–80.
- [34] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: a review of methods and applications, *AI Open* 1 (2020) 57–81.
- [35] B. Li, X. Li, Z. Zhang, F. Wu, Spatio-temporal graph routing for skeleton-based action recognition, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, 2019, pp. 8561–8568.
- [36] L. Shi, Y. Zhang, J. Cheng, H. Lu, Skeleton-based action recognition with directed graph neural networks, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7912–7921.
- [37] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: *Proceedings of International Conference on Learning Representations*, 2017.
- [38] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: *Proceedings of International Conference on Learning Representations*, 2018.
- [39] Y. Chen, C. Liu, B.E. Shi, M. Liu, Robot navigation in crowds by graph convolutional networks with attention learned from human gaze, *IEEE Rob. Autom. Lett.* 5 (2) (2020) 2754–2761.
- [40] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.
- [41] S. Pellegrini, A. Ess, L. Van Gool, Improving data association by joint modeling of pedestrian trajectories and groupings, in: *Proceedings of European Conference on Computer Vision*, Springer, 2010, pp. 452–465.
- [42] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.
- [43] S. Yan, Y. Xiong, D. Lin, Spatial temporal graph convolutional networks for skeleton-based action recognition, in: *Proceedings of Thirty-second AAAI Conference on Artificial Intelligence*, 2018.
- [44] A. Lerner, Y. Chrysanthou, D. Lischinski, Crowds by example, in: *Computer graphics forum*, Vol. 26, Wiley Online Library, 2007, pp. 655–664.
- [45] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, S. Savarese, SoPhie: an attentive GAN for predicting paths compliant to social and physical constraints, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1349–1358.
- [46] H. Sun, Z. Zhao, Z. He, Reciprocal learning networks for human trajectory prediction, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7416–7425.
- [47] C. Tao, Q. Jiang, L. Duan, P. Luo, Dynamic and static context-aware LSTM for multi-agent motion prediction, in: *Proceedings of European Conference on Computer Vision*, Springer, 2020, pp. 547–563.
- [48] H. Bi, R. Zhang, T. Mao, Z. Deng, Z. Wang, How can i see my future? FvTraj: using first-person view for pedestrian trajectory prediction, in: *Proceedings of European Conference on Computer Vision*, Springer, 2020, pp. 576–593.
- [49] H. Sun, Z. Zhao, Z. Yin, Z. He, Reciprocal twin networks for pedestrian motion learning and future path prediction, *IEEE Trans. Circuits Syst. Video Technol.* (2021).

Xingchen Zhang is a Marie Skłodowska-Curie Individual Fellow at the Personal Robotics Lab, Department of Electrical and Electronic Engineering, Imperial College London. Prior to this, he was a Teaching Fellow and Research Associate at the same department. His main research interests include human motion prediction, human intention prediction, image fusion, computer vision, and deep learning.

Panagiotis Angeloudis is a Reader and the Head of the Transport Systems and Logistics Laboratory, Imperial College London. He was recently appointed by the U.K. Department for Transport to the Expert Panel for Maritime 2050. He was a member of the U.K. Government Office of Science Future of Mobility Review Team. His research focuses on the study of networks, optimisation methods and multi-agent systems, as well as their applications in autonomous transport systems, urban infrastructure and logistics.

Yiannis Demiris is a Professor with the Department of Electrical and Electronic Engineering, Imperial College London, London, U.K., where he is the Royal Academy of Engineering Chair in Emerging Technologies, and the Head of the Personal Robotics Laboratory. His current research interests include human-robot interaction, machine learning, user modeling, and assistive robotics. He is a Fellow of IET.