THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 25, Number 4/2024, pp. 335–346
DOI: 10.59277/PRA-SER.A.25.4.10

AN ACCURATE SPATIAL TEMPORAL GRAPH ATTENTION NETWORK FOR PEDESTRIAN TRAJECTORY PREDICTION

Yanbo ZHANG, Liying ZHENG

Harbin Engineering University, School of Computer Science and Technology, Harbin, 150001, China
Corresponding author: Liying ZHENG, E-mail: zhengliying@hrbeu.edu.cn

Abstract. Pedestrian trajectory prediction has broad applications to target tracking and autonomous driving. Although current research has gradually shifted from handcrafted-based approaches to deep learning-based approaches, existing predictors sometimes produce wrong future trajectories, and their performance is still unsatisfied. This paper improves the Spatial Temporal Graph ATtention Network (STGAT) for trajectory prediction by adding two Multi-Layer Perceptron (MLP) modules and three Gated Recurrent Units (GRUs) to the original predictor. Specifically, the model uses an MLP to process the position and velocity of a pedestrian to obtain high-dimensional embedding features. Then, to improve prediction accuracy, the model uses a GRU and a Long Short-Term Memory module (LSTM), i.e. GRU-LSTM, to obtain the motion features of the individual pedestrian. Next, the interaction information between the target pedestrian and his/her neighbors is captured by a Graph ATtention (GAT) module. Here, to reduce the redundant information in the GAT, another MLP is used to map the high-dimensional features to low-dimensional ones. Next, the second GRU-LSTM further encodes the interaction features from the GAT. Finally, the third GRU-LSTM serves as a decoder to give the future trajectory of the target pedestrian. To generate multiple socially acceptable prediction results, the model adopts the variety loss. Test results on the public ETH and UCY datasets illustrate that the proposed model outperforms the state-of-the-art predictors such as Social-LSTM, Social-Attention, CIDNN, and etc. Moreover, the ablation tests show that by using the MLPs and GRUs, the ADE and FDE of the model are lower than the STGAT and Social-GAN, further proving the benefits of these new adding modules to the predictor.

Keywords: pedestrian trajectory prediction, multi-layer perceptron, gated recurrent unit, long short-term memory, graph attention network.

1. INTRODUCTION

Pedestrian trajectory prediction is widely used in military and civilian applications, such as automatic driving, intelligent navigation and human-computer interaction. Predicting a pedestrian trajectory means generating his/her future locations based on the past ones. Pedestrians' movements are highly complex and stochastic since many factors influence people's judgment on the following target location, such as surrounding buildings, destinations, and other pedestrians. Despite the significant progress in crowd trajectory prediction in the past few years, pedestrian trajectory prediction still needs to be improved and attracts many researchers.

Traditional hand-crafted-feature based approaches only consider the pedestrians themselves, ignoring their interactions. Therefore, in recent years, Recurrent Neural Network (RNN) [1] techniques have been introduced to trajectory prediction. Though being proven effective, using RNN and their variants [2–3] alone cannot capture pedestrian interactions. Then, Graph Neural Network (GNN) [4] and their variants [5–6] are used for modeling pedestrian interactions with the graph structure, getting more accurate predictions. Currently, there are two main types of pedestrian trajectory prediction approaches based on GNN, i.e., the Graph Convolutional Network (GCN) [5] and the Graph ATtention Network (GAT) [6]. GCN fuses information based on neighborhood aggregation and cannot capture the influence of surrounding pedestrians on the target one. In contrast, GAT proposes an attention mechanism to capture the contribution of each neighbor node to the target pedestrian, resulting in good prediction performance. By introducing Multi-Layer Perceptron modules (MLPs) and Gated Recurrent Units (GRUs) to the original predictor, this paper proposes an improved

Spatial Temporal Graph ATtention network (STGAT) [7] for pedestrian trajectory prediction. This paper's contributions are summarized as below:

- Introducing GRUs to the STGAT to improve the prediction accuracy of the future trajectory of a target pedestrian;
- Using an MLP to obtain high-dimensional embedding features of the position and velocity;
- Using an MLP to reduce the redundant information in the GAT module;
- Using the variety loss for training the model to generate multiple socially-compliant prediction results.

The remainder of this paper is as follows. Section 2 introduces existing approaches and their limitations. Section 3 briefly introduces the STGAT model. Section 4 describes the proposed pedestrian trajectory predictor in detail. Section 5 presents test results and analysis, and Section 6 gives conclusions and future work.

2. RELATED WORK

2.1. RNN-based trajectory prediction

Recently, the data-driven RNN and its variants [2–3], including Long Short-term Memory (LSTM) [2] and Gated Recurrent Unit (GRU) [3] with an encoder-decoder, have been widely used in trajectory prediction. Wu *et al.* [8] proposed two RNNs for trajectory prediction, which simultaneously address the limitations of topology on prediction. Shibata *et al.* [9] introduced reinforcement learning to RNNs to predict decisions for discrete and continuous motion targets. Moleg *et al.* [10] designed a spatial-temporal feature-based RNN to predict the future motion state of pedestrians by fusing spatial and temporal associations in pedestrian trajectory data. Tang *et al.* [11] combined the history feature fusion attention with LSTM to improve the model's predictive performance when dealing with long sequences.

2.2. MLP-based trajectory prediction

Multi-Layer Perceptron (MLP) [12] is a simple model commonly used for data preprocessing in pedestrian trajectory prediction tasks. For example, Alahi *et al.* [13] proposed a Social LSTM (S-LSTM), which uses an MLP for preprocessing and an LSTM for matching physical attributes and social norms. Then some extensions to the S-LSTM emerged. Pfeiffer *et al.* [14] added intergroup interaction to S-LSTM and clustered trajectories with similar motion trends. Bartoli *et al.* [15] considered the influence of static environment on pedestrians based on the S-LSTM model. Xu *et al.* [16] proposed a collision-free LSTM model, redefined the concept of "neighbors" in the S-LSTM, and converted right-angle coordinates to polar coordinates. Choi *et al.* [17] mapped the location of each pedestrian to a high-dimensional feature space and then modeled the motion of all pedestrians using LSTM.

2.3. GRU-LSTM-based trajectory prediction

RNN and their variants, such as LSTM and GRU, are popular trajectory predictors, too. LSTM extends storage capacity to preserve and process previous information and thus performs well for long-term prediction. GRU has fewer parameters and is faster and more accurate. Thus, it is popular to use GRU-LSTM for time series prediction, i.e., utilizing GRU for prediction first and then LSTM for improving long-term predictions. Liu *et al.* [18] proposed a regularized GRU-LSTM model for predicting stock closing prices. Islam *et al.* [19] used the hybrid GRU-LSTM to forecast foreign exchange rates, getting better results than traditional approaches. Kianimoqadam *et al.* [20] introduced GRU-LSTM for distance-ordered sequential data to predict view factors of the particle and face neighbors of an emitting particle. Sari Y *et al.* [21] explored the impact of the number of input parameters on the performance of the hybrid GRU-LSTM in predicting air temperature.

2.4. GNN-based trajectory prediction

Unlike RNNs that use common aggregation operations to process spatial-temporal features, GNNs rely on information transferred between nodes to capture dependencies in a graph. There are two basic GNN models for pedestrian trajectory prediction, i.e., GCN and GAT. Dan *et al.* [22] extracted features of pedestrians with a GCN by encoding each pedestrian as a node, and thus the interrelationships between pedestrians can be

extracted by the graph structure. Yu *et al.* [23] proposed a spatial-temporal association model by combining a Transformer and a GCN. In their approaches, Yu *et al.* used the GCN to model the movement trend of pedestrians in the group interaction scene.

Though GCN-based approaches use neighborhood aggregation for fusing information, they cannot capture the influence of surrounding pedestrians on the target one. To solve this problem, GAT-based models adopt an attention mechanism to identify the contribution of each adjacent node, achieving good test results. Huang *et al.* [7] tried different attentions based on a GAT to achieve effective weighted information transfer between nodes. Kosaraju *et al.* [24] proposed a latent spatial encoder that encodes pedestrian interactions with a GAT and uses a modified Bicycle-GAN [25] to generate multiple predicted trajectories that meet social criteria.

Similar to STGAT [7], in the proposed approach, pedestrian interactions are represented as a graph, with nodes designating people and edges indicating interactions. An attention mechanism is used to obtain corresponding weights for each node to reduce the redundant information in the high-dimensional space; this paper first attempts to combine GAT with MLP to determine the influence between people.

3. STGAT

Pedestrian trajectory prediction involves forecasting the future movement trajectories of pedestrians by analyzing historical data and surrounding environments. This process finds extensive application in autonomous driving, robot navigation, intelligent transportation, and video surveillance systems. The pioneering work on pedestrian trajectory prediction using STGAT was conducted by Huang *et al.* [7]. STGAT is designed to establish the relationships between pedestrians in both temporal and spatial contests through GAT and LSTM.

As shown in Fig. 1, the encoder of the STGAT adopts two LSTMs (M-LSTM for individual motion encoding and G-LSTM for the graph encoding) and a GAT to encode spatial-temporal social interaction information. The decoder of the STGAT contains an additional LSTM (D-LSTM for decoding). Besides, each LSTM is followed by a Fully Connected (FC) layer. First, the M-LSTM processes the positions and velocities of a pedestrian to obtain his/her individual motion characteristics. Then, the GAT gets the spatial-temporal interaction information that is further processed by the G-LSTM. Next, features characterizing individual pedestrian motion are input to an FC. So do the features characterizing spatial-temporal interactions. After concatenating with noise, the fused features are decoded by D-LSTM followed by another FC to get the relative predicted position.

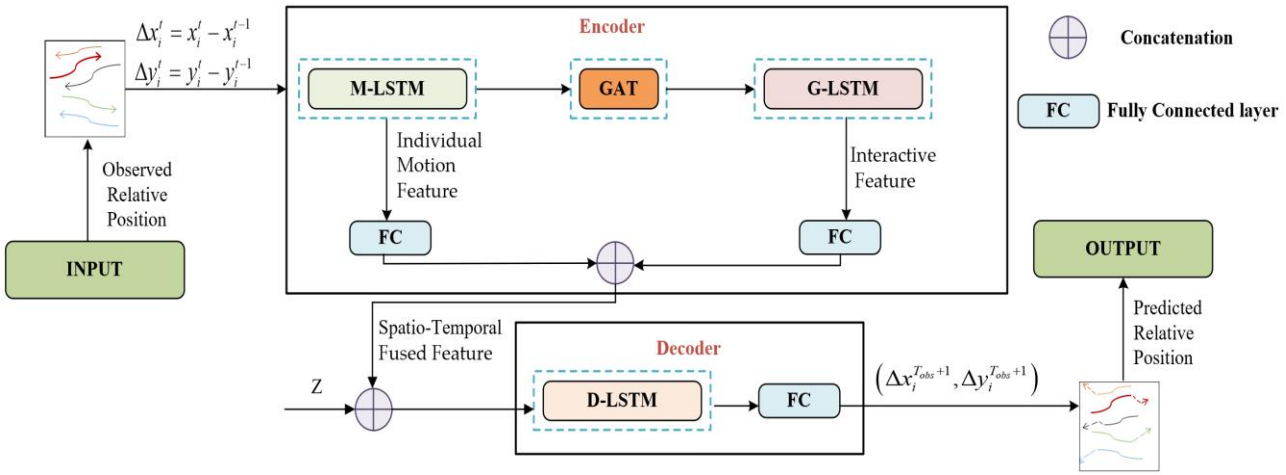

Fig. 1 – Structure of STGAT.

4. APPROACH

Existing studies show that GRU benefits the accuracy of a predictor [18–21]. Thus, as shown in Fig. 2, the model adds a GRU to each LSTM module in the original STGAT to improve its accuracy. By convention, the combined module is termed as GRU-LSTM. Moreover, to obtain more motion details of a pedestrian, the model adds an MLP to embed the observed positions and velocities of the pedestrian to high-dimensional features. Finally, to reduce the redundancy in the GAT of the original STGAT, the model appends another MLP to the GAT.

Specifically, the encoder consists of two GRU-LSTMs (i.e., M-GRU-LSTM and G- GRU-LSTM), two MLPs with the same structure, and a GAT. Similar to STGAT, each GRU-LSTM in the model is followed by an FC layer. Here, M-GRU-LSTM and G-GRU-LSTM encode the individual motion and their interaction information, respectively. The two MLPs are used for feature expanding and feature reducing. The GAT is for capturing the spatial-temporal social relations between pedestrians. The decoder is similar to that in STGAT, except that the model is used the GRU-LSTM rather than the LSTM.

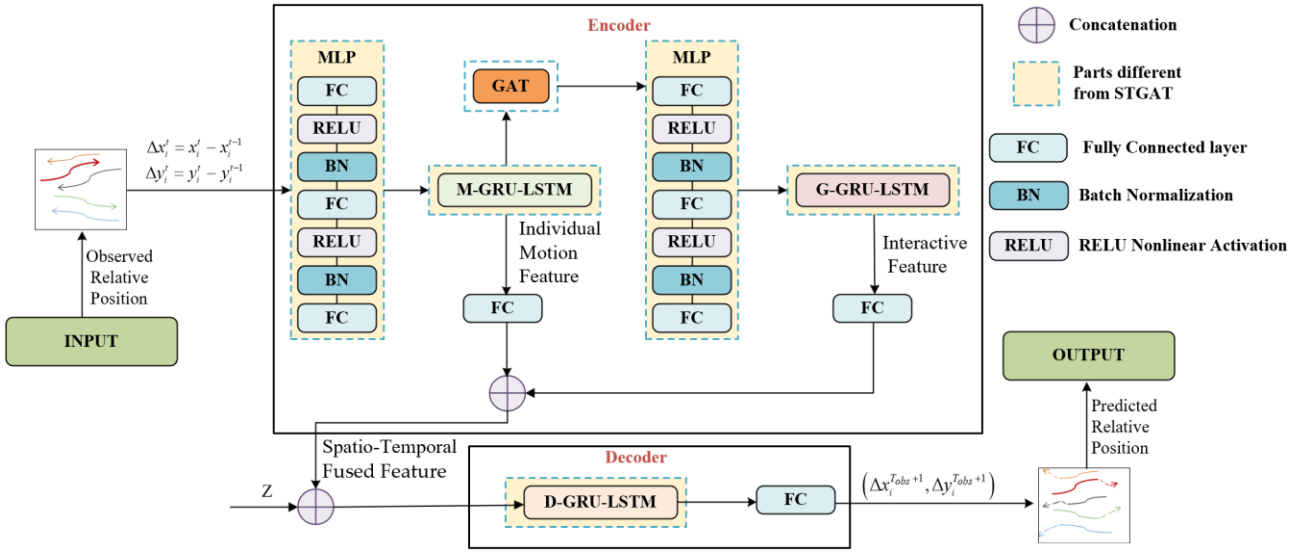

Fig. 2 – The structure diagram of the proposed approach.

4.1. MLP modules

As shown in Fig. 2, to preprocess the position and velocity information of a pedestrian, the model uses an MLP consisting of three Fully Connected (FC) [26] layers, two RELU nonlinear activation [27] layers and two Batch Normalization (BN) [28] layers. This MLP maps the relative replacement $(\Delta \boldsymbol{x}_i^t, \Delta \boldsymbol{y}_i^t)$ of i^{th} the pedestrian at time instant t to a high-dimensional embedding features $\boldsymbol{e}_i^t$ with (1)–(5)

$$\Delta \boldsymbol{x}_i^t = \phi_{\text{relu}}\left(\boldsymbol{W}_{\Delta x} \Delta \boldsymbol{x}_i^t + \boldsymbol{b}_{\Delta x}\right) \tag{1}$$

$$\Delta \boldsymbol{x}_i^t = \phi_{bn}\left(\Delta \boldsymbol{x}_i^t; \boldsymbol{W}_{\Delta x}\right) \tag{2}$$

$$\Delta \boldsymbol{y}_i^t = \phi_{\text{relu}}\left(\boldsymbol{W}_{\Delta y} \Delta \boldsymbol{y}_i^t + \boldsymbol{b}_{\Delta y}\right) \tag{3}$$

$$\Delta \boldsymbol{y}_i^t = \phi_{bn}\left(\Delta \boldsymbol{y}_i^t; \boldsymbol{W}_{\Delta y}\right) \tag{4}$$

$$\boldsymbol{e}_i^t = \phi_{fc}\left(\Delta \boldsymbol{x}_i^t, \Delta \boldsymbol{y}_i^t; \boldsymbol{W}_{\Delta y}\right) \tag{5}$$

where ϕ_{relu}, ϕ_{bn}, ϕ_{fc} are RELU nonlinear activation, batch normalization and fully connected embedding function, respectively; $\boldsymbol{W}_{\Delta x}$, $\boldsymbol{W}_{\Delta y}$, $\boldsymbol{W}_m$ are the weights, $\boldsymbol{b}_{\Delta x}$, $\boldsymbol{b}_{\Delta y}$ are the bias.

As illustrated in Fig. 2, following the GAT module, an additional MLP is appended to map the output features of the GAT to a lower-dimensional space. This step effectively reduces feature redundancy. The structure of this MLP is designed to be identical to that of the first one.

4.2. GAT module

Similar to STGAT, the model uses the self-attention approach. Each node feature in the graph is calculated based on the features of its neighbors. The model uses two graph attention layers in GAT. During the observation process, the individual motion feature vector $\boldsymbol{m}_i^t$ is fed into the graph attention layer. With the attention coefficients computed by (6), one can get the output $\hat{\boldsymbol{m}}_i^t$ of a single graph attention layer for the i^{th} node at time t with (7) that contains the spatial influence of the i^{th} pedestrian from other pedestrians at time t.

$$\boldsymbol{\alpha}_{ij}^t = \frac{\exp\left(LeakyReLU\left(\boldsymbol{a}^T\left[\boldsymbol{W}\boldsymbol{m}_i^t \oplus \boldsymbol{W}\boldsymbol{m}_j^t\right]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(LeakyReLU\left(\boldsymbol{a}^T\left[\boldsymbol{W}\boldsymbol{m}_i^t \oplus \boldsymbol{W}\boldsymbol{m}_k^t\right]\right)\right)} \tag{6}$$

$$\hat{\boldsymbol{m}}_i^t = \sigma\left(\sum_{j\in\mathcal{N}_i}\boldsymbol{\alpha}_{ij}^t\boldsymbol{W}\boldsymbol{m}_j^t\right) \tag{7}$$

where $\oplus$ denotes the concatenation operation, and T means transposition. $\boldsymbol{\alpha}_{ij}^t$ refers to the attention coefficient of node j towards node i at time t, and $\mathcal{N}_i$ denotes the neighbors of node i in the graph. A shared linear transformation weight matrix $\boldsymbol{W} \in R^{F'\times F}$ is applied to each node with F and F' being the dimensions of $\hat{\boldsymbol{m}}_i^t$ and the output, respectively. The weight $\boldsymbol{a} \in R^{2F'}$ belongs to a single-layer feed-forward neural network and normalized using a LeakyReLU softmax function.

4.3. GRU-LSTM Modules

As shown in Fig. 3, to improve the prediction accuracy, the model adds a GRU module to each LSTM of the original STGAT, getting three GRU-LSTM modules, i.e., M-GRU-LSTM, G-GRU-LSTM, and D-GRU-LSTM. Their structures are similar.

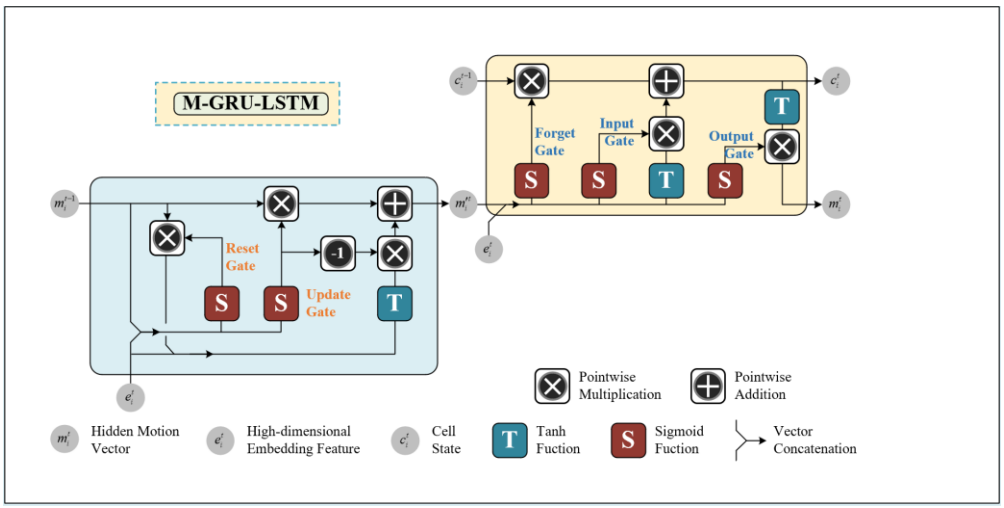

Fig. 3 – The M-GRU-LSTM module (the structures of G-GRU-LSTM module and D-GRU-LSTM module are similar).

As shown in Fig. 3, M-GRU-LSTM contains a one-layer GRU and a one-layer LSTM. First, as given in (8)–(11), the GRU module estimates the transient motion vector $\boldsymbol{m}'^t_i$ according to the high-dimensional embedding features $\boldsymbol{e}_i^t$ and the previous hidden motion vector $\boldsymbol{m}_i^{t-1}$. Then, with $\boldsymbol{m}'^t_i$ at hand, the LSTM module computes the hidden motion feature vector $\boldsymbol{m}_i^t$ with (12)–(16).

$$\boldsymbol{r}_i^t = \text{Sig}\left(\boldsymbol{W}_{er}\boldsymbol{e}_i^t + \boldsymbol{W}_{mr}\boldsymbol{m}_i^{t-1} + \boldsymbol{b}_r\right) \tag{8}$$

$$\boldsymbol{z}_i^t = \text{Sig}\left(\boldsymbol{W}_{ez}\boldsymbol{e}_i^t + \boldsymbol{W}_{mz}\boldsymbol{m}_i^{t-1} + \boldsymbol{b}_z\right) \tag{9}$$

$$\hat{\boldsymbol{m}}'^t_i = \text{Tanh}\left(\boldsymbol{W}_{ec}\boldsymbol{e}_i^t + \boldsymbol{W}_{mc}\left(\boldsymbol{r}_i^t \times \boldsymbol{m}_i^{t-1}\right) + \boldsymbol{b}_c\right) \tag{10}$$

$$\boldsymbol{m}'^t_i = \left(1 - \boldsymbol{z}_i^t\right) \times \boldsymbol{m}_i^{t-1} + \boldsymbol{z}_i^t \times \hat{\boldsymbol{m}}'^t_i \tag{11}$$

where $\boldsymbol{m}_i^{t-1}$ represents the hidden motion feature of the target pedestrian at time $t-1$. $\mathrm{Sig}(\cdot)$ and $\mathrm{Tanh}(\cdot)$ are the sigmoid function and tanh function, respectively; $\boldsymbol{W}_{er}, \boldsymbol{W}_{mr}$ and $\boldsymbol{b}_r$ are the weights and the bias of the reset gate; $\boldsymbol{W}_{ez}, \boldsymbol{W}_{mz}$ and $\boldsymbol{b}_z$ are the weights and the bias of the update gate. and $\widehat{\boldsymbol{m}}_i^{'t}$ is a candidate memory cell; $\boldsymbol{W}_{ec}, \boldsymbol{W}_{mc}$ and $\boldsymbol{b}_c$ are the weights and the bias of the update gate; $\boldsymbol{m}_i^{t-1}$ and $\boldsymbol{m}_i^{'t}$ are hidden motion vector and transient motion vector.

$$\boldsymbol{f}_i^t = \mathrm{Sig}\left(\boldsymbol{W}_{ef}\boldsymbol{e}_i^t + \boldsymbol{W}_{mf}\boldsymbol{m}_i^{'t} + \boldsymbol{b}_f\right) \tag{12}$$

$$\boldsymbol{i}_i^t = \mathrm{Sig}\left(\boldsymbol{W}_{ei}\boldsymbol{e}_i^t + \boldsymbol{W}_{mi}\boldsymbol{m}_i^{'t} + \boldsymbol{b}_i\right) \tag{13}$$

$$\hat{\boldsymbol{c}}_i^t = \mathrm{Tanh}\left(\boldsymbol{W}_{e\hat{c}}\boldsymbol{e}_i^t + \boldsymbol{W}_{m\hat{c}}\boldsymbol{m}_i^{'t} + \boldsymbol{b}_{\hat{c}}\right) \tag{14}$$

$$\boldsymbol{c}_i^t = \boldsymbol{f}_i^t\boldsymbol{c}_{t-1} + \boldsymbol{i}_i^t \otimes \hat{\boldsymbol{c}}_i^t \tag{15}$$

$$\boldsymbol{m}_i^t = \mathrm{Sig}\left(\boldsymbol{W}_{eo}\boldsymbol{e}_i^t + \boldsymbol{W}_{mo}\boldsymbol{m}_i^{'t} + \boldsymbol{b}_o\right) \otimes \mathrm{Tanh}\boldsymbol{c}_i^t \tag{16}$$

where $\boldsymbol{f}_i^t$ and $\boldsymbol{i}_i^t$ are the outputs of forget gate and input gate, and $\boldsymbol{W}_{ef}, \boldsymbol{W}_{mf}, \boldsymbol{W}_{ei}, \boldsymbol{W}_{mi}$ are their weights, $\boldsymbol{b}_f$ and $\boldsymbol{b}_i$ are their biases, respectively; $\boldsymbol{c}_i^t$ and $\hat{\boldsymbol{c}}_i^t$ are the memory cell and the candidate memory cell; $\boldsymbol{W}_{e\hat{c}}, \boldsymbol{W}_{m\hat{c}}$ and $\boldsymbol{b}_{\hat{c}}$ are the weights and bias of the update gate; $\boldsymbol{W}_{eo}, \boldsymbol{W}_{mo}$ and $\boldsymbol{b}_o$ are the weights and bias of the output gate.

Similar to M-GRU-LSTM, the outputs of G-GRU-LSTM and D-GRU-LSTM can be obtained with (8)–(16) according to their own input feature vectors.

4.4. Loss function

As in [29], the variety loss is used to train the model. First, for each future trajectory predicted, the model generates *K* predicted trajectories by randomly adding noise that samples from the standard normal distribution. Then, the loss of the model is computed using L2 norm of the K nearest trajectories to the ground-truth. The above mentioned procedure is repeated *N* times, and the loss is given by (17).

$$\boldsymbol{L} = \min_{N} \min_{n=\{1,2,\ldots K\}} \left\| \boldsymbol{Y}_i - \widehat{\boldsymbol{Y}}_i^{(n)} \right\|_2 \tag{17}$$

where $\boldsymbol{Y}_i$ is the ground-truth trajectory of the i^{th} pedestrian, and $\widehat{\boldsymbol{Y}}_i^{(n)}$ is the n^{th} random trajectory, $\|\cdot\|_2$ is the $L2$ norm.

4.5. Algorithm process

Inspired by the STGAT model, this paper presents the design and training of an end-to-end encoder-decoder model. The encoder comprises five key components:

- High-dimensional embedding features are extracted using a Multi-Layer Perceptron (MLP) model;
- A GRU-LSTM-based model is employed to encode pedestrian trajectories;
- A GAT-based module is used to model spatial interactions between pedestrians;
- The model is trained using a variety of loss functions to generate multiple prediction outcomes that are socially compliant;
- To reduce redundant information, another MLP model with the same structure is used to map high-dimensional features to low-dimensional representations, resulting in precise;
- A GRU-LSTM-based model is utilized to capture the temporal correlations in interactions.

In the decoder component, a GRU-LSTM model is utilized for decoding, and a diversity loss function is introduced to encourage the generation of diverse output samples. The effectiveness of this approach is confirmed through validation. For this study, the Adam optimizer is configured with a learning rate of 0.01 and a batch size of 64. Algorithm 1 presents the overall algorithmic flow.

Table 1 details the model's structure and dimensions; the GRU-LSTMs have one GRU layer and one LSTM layer. $\boldsymbol{\phi}_{relu}$ in Eq. 1 and Eq. 3 with two ReLU activation functions in the first MLP model and the number of hidden nodes is 16, 16; the number of hidden nodes is 16, 8 in the second MLP model; the dimension of e_i^t in Eq. 5 set to 16; the dimension of m_i^t in Eq.6 set to 32; the shapes of $\boldsymbol{W}$ in the first and second layers are 16×16 and 16×8, respectively; the dimensions of $\boldsymbol{a}$ are 32 and 64 in the first and second layers, respectively; the dimensions of $\widehat{\boldsymbol{m}}_i^t$ and $\boldsymbol{m}_i^{'t}$ in the three GRU-LSTMs are 16, 8, 8.

Table 1

Architecture details of the model

Layers	Structure
The 1st MLP	→FC(2,16)→BN1d→ReLU(16)→FC(16,16) →BN1d→ReLU(16)→FC(16,16)
The 2nd MLP	→FC(16,32)→BN1d→ReLU(16)→FC(32,16) →BN1d→ReLU(8)→FC(16,8)
The 1st GAT layer	BatchMultiHeadGraphAttention (4→32→16)
The 2nd GAT layer	BatchMultiHeadGraphAttention (1→64→8)
M-GRU-LSTM	→GRUCell(16,32)→LSTMCell(16,32)
G-GRU-LSTM	→GRUCell(8,8)→LSTMCell(8,8)
D-GRU-LSTM	→GRUCell(8,48)→LSTMCell(8,48)
The 1st FC layer	→FC(32,2)
The 2nd FC layer	→FC(40,2)
The 3rd FC	→FC(48,2)

Algorithm 1 summarizes the main steps of the proposed model.

Algorithm 1 Proposed model: iteration at frame *t*.

Inputs: the relative replacement $(\Delta \boldsymbol{x}_i^t, \Delta \boldsymbol{y}_i^t)$ of i^{th} the pedestrian at time instant *t*.

Outputs: the predicted relative replacement $\left(\Delta \boldsymbol{x}_i^{T_{obs}+1}, \Delta \boldsymbol{y}_i^{T_{obs}+1}\right)$ of i^{th} the pedestrian at time instant *t*.

Level 1: modeling the movement of pedestrians themselves for high-dimensional features:

Using an MLP to high-dimensional embedding features $\boldsymbol{e}_i^t$ using (1)–(5).

Level 2: pedestrian interaction GAT module:

for each vector $\widehat{\boldsymbol{m}}_i^t$ **do**:

The individual motion feature vector $\boldsymbol{m}_i^t$ is fed into the graph attention layer.

The attention coefficients are computed by (6).

The output $\widehat{\boldsymbol{m}}_i^t$ of a single graph attention layer for the i^{th} node at time *t* with (7).

Level 3: GRU-LSTM module:(M-GRU-LSTM, G-GRU-LSTM and D-GRU-LSTM).

Adding a GRU module to each LSTM of the original STGAT.

1.The GRU module estimates the transient motion vector $\boldsymbol{m}'^t_i$ according to the high-dimensional embedding features $\boldsymbol{e}_i^t$ with (8)–(11).

2.The previous hidden motion vector $\boldsymbol{m}_i^{t-1}$. Then, with $\boldsymbol{m}'^t_i$ at hand, the LSTM module computes the hidden motion feature vector $\boldsymbol{m}_i^t$ with (12)–(16).

3. Decoding by G-GRU-LSTM to get the predicted relative replacement $\left(\Delta \boldsymbol{x}_i^{T_{obs}+1}, \Delta \boldsymbol{y}_i^{T_{obs}+1}\right)$.

Level 4: the variety loss:

The Adam optimizer is used to help the model adjust the parameters.

for each *i*:

The model generates *K* predicted trajectories by randomly adding noise that samples from the standard normal distribution. Then, the model selects the trajectory with the smallest gap from the true value as the model output to calculate the loss as in (17).

end.

5. TESTS AND ANALYSIS

The architectural details of the proposed model are listed in Table 1. Using Adam optimizer [30] with a learning rate of 0.01 and a batch size of 64 for 400 epochs. Adam is an adaptive learning rate optimization algorithm that effectively accelerates convergence and enhances model stability. This paper performs well in handling large-scale datasets and complex deep-learning tasks. In the Pytorch framework on Ubuntu 20.04, the STGAT and proposed model are equipped with an NVIDIA 2080Ti (11GB memory) and an i7 10 700 K processor. The model was validated on two benchmark datasets, i.e., ETH [31]and UCY [32]. All data are converted to the world coordinate system and then interpolated at every 0.4 seconds for the ETH & UCY test

setting. During the prediction, the model generates future 12-time-step trajectory of a pedestrian from the past 8-time-step observations. The model adopts the “leave-one-out” [33] approach to train and test the proposed model, the five datasets are partitioned, with four used for training and one for testing. After training, the model's performance is evaluated on the test sample, and the results are recorded. This approach maximizes the dataset's utility for training and critically assesses the model's generalization capabilities.

Similar to STGAT, the model employs two metrics to evaluate the prediction accuracy of a predictor:

• Average displacement error (ADE): Root-mean-square error (RMSE) of all estimated positions in predicted and actual trajectories;

• Final displacement error (FDE): The distance between the predicted and actual final destination.

Following STGAT, with three settings of and in (13), the model gets three versions, referred to as OURS-Kv-N. Here, that means no loss of variety. The results listed in Table 2 show that the setting outperforms others for all datasets. Therefore, hereafter, the model uses OURS-20v-20 as the final predictor. In all tests, this paper reported the average result over three runs.

Table 2

AED and FDE of the model with three settings (the best results are in bold)

Approaches	ADE/FDE					
	ZARA2	ZARA1	UNIV	HOTEL	ETH	AVG
OURS-1V-1	0.37/0.78	0.41/0.87	0.55/1.17	0.44/0.93	0.92/1.85	0.54/1.12
OURS-1V-20	0.33/0.69	0.37/0.79	0.54/1.16	0.37/0.78	0.87/1.13	0.49/1.03
OURS-20V-20	**0.28/0.57**	**0.32/0.65**	**0.53/1.15**	**0.32/0.59**	**0.66/1.11**	**0.42/0.81**

5.1. Comparison with the state-of-the-art models

This paper compares the proposed model with 10 popular predictors, the detailed structure of the model is shown in Table 3, which describes the structure of the same and different modules and the structure of other models on the surface.

Table 3

The detailed structure of 10 popular predictors (including the same and different architectures)

Approaches	The Same Architectures	The Different Architectures
Social-LSTM [13]	Multiple LSTM Models;	A Social Pooling Model.
Social-GAN-20V-20 [29]	Two LSTM Models; Two FC Models Two RELU Models.	A Global Pooling Model.
Social-Attention [34]	Two RNN Models; An Attention Model.	A Spatial-Temporal Graph.
CIDNN [35]	Two LSTM Models; A FC Model.	A Spatial Affinity Model.
BiGAN [24]	Three LSTM Models Per Component; A Bicycle-GAN Model; Five MLP Models.	A Reversible Mapping.
SAGCN [37]	A GCN Model; A TCN Model.	A Graph Adjacency Matrix
CoMoGCN [38]	Two FC+ LSTM Models; Two FC Models; Two GCN Models.	Variational Autoencoders.
RSBG [39]	An LSTM Model; A GCN Model; A CNN Model.	A Bi-LSTM Model; A Recursive Social Behavior Graph.
BR-GAN [36]	A GAN Model; A Geographical Attention Model; A Social Attention Model; A Behavior Attention Model.	VGG-16 Net; A Novel Behavior Recognition Model.
STGAT-20V-20 [7]	Two FC Models; Three LSTM Models.	A Spatial-Temporal GAT Model.
OURS-20V-20	Two FC Models; Two RELU Models; A Spatial-Temporal GAT Model.	Two BN Models; A FC Model; Three GRU-LSTM Models.

Table 4 lists the test results of the model, STGAT, and Social-GAN. All of these three predictors adopt variety loss. Compared to Social-GAN, the average ADE and FDE are reduced by 38.1% and 43.9% for the model, respectively. Similarly, compared to STGAT, these two metrics are reduced by 7.1% and 6.1%.

Table 5 gives the results of the model and the other 8 popular ones. Among the compared 8 models, CoMoGCN and RSBG show the latest performance regarding the average ADE and FDE. Compared with these two predictors, the approach decreases the average ADE by 7.14% and 14.3%, respectively, while decreasing the average FDE by 11.0% and 20.7%. Besides, Table 4 shows that the model performs best on all ADE sub-datasets and four FDE sub-datasets. On average, the model is the best among all the 10 approaches.

5.2. Ablation

5.2.1. Evaluation of MLPs and GRU-LSTMs

Compared to the original STGAT, the model adds two MLPs and substitutes GRU- LSTM for LSTM. To demonstrate these new adding modules, the model evaluates five variants, i.e., the original STGAT(STGAT-20V-20), the model with GRU-LSTM, the model with GRU-LSTM and the first MLP (GRU-LSTM-MLP1), the model with GRU- LSTM and the second LSTM(GRU-LSTM-MLP2), and the model with all new adding ones OURS (OURS-20V-20). The results are listed in Table 5. It is evident that, compared to STGAT, GRU-LSTM-MLP2 achieves an average ADE reduction of 2.3% and an average FDE reduction of 1.2%. A similar trend is observed for GRU-LSTM-MLP1. For GRU-LSTM, the average ADE and FDE are reduced by 4.7% and 1.2%, respectively. The results indicate that including all newly added modules improves prediction accuracy. Furthermore, the OURS model (OURS-20V-20) exhibits the lowest ADE across all sub-datasets and achieves the lowest FDE on three sub-datasets, demonstrating the best overall performance in average ADE and FDE.

Table 4

The ADE and FDE of the model, SGAN, and STGAT (the best results are in bold)

Approaches	ADE/FDE					
	ZARA2	ZARA1	UNIV	HOTEL	ETH	AVG
Social-GAN-20V-20 [29]	0.42/0.84	0.34/0.69	0.60/1.26	0.72/1.61	0.81/1.52	0.58/1.18
STGAT-20V-20 [7]	0.30/0.61	0.34/0.68	0.56/1.20	0.33/0.62	0.71/1.26	0.45/0.87
OURS-20V-20	**0.28/0.57**	**0.32/0.65**	**0.53/1.15**	**0.32/0.59**	**0.66/1.11**	**0.42/0.81**

Table 5

The ADE and FDE of the model and other 8 popular ones (the best results are in bold)

Approaches	ADE/FDE					
	ZARA2	ZARA1	UNIV	HOTEL	ETH	AVG
Social-LSTM [13]	0.56/1.17	0.47/1.00	0.67/1.40	0.79/1.76	1.09/2.35	0.72/1.54
Social-Attention [34]	0.88/1.75	1.01/2.17	1.25/2.54	2.51/2.91	1.39/2.39	1.41/2.35
CIDNN [35]	0.51/1.07	0.50/1.04	0.90/1.86	1.31/2.36	1.25/2.23	0.89/1.73
BiGAN [24]	0.49/0.88	**0.32/0.65**	0.55/1.34	0.54/1.12	0.72/1.47	0.52/1.09
SAGCN [37]	0.32/0.70	0.41/0.89	0.57/1.19	0.41/0.83	0.90/1.96	0.52/1.11
CoMoGCN [38]	0.31/0.67	0.34/0.71	**0.53**/1.16	0.37/0.75	0.70/1.26	0.45/0.91
RSBG [39]	0.30/0.65	0.40/0.86	0.59/1.25	0.33/0.64	0.80/1.53	0.48/0.99
BR-GAN [36]	0.35/0.72	0.35/0.71	**0.53/1.07**	0.55/1.13	0.73/1.37	0.50/1.00
OURS-20V-20	**0.28/0.57**	**0.32/0.65**	**0.53**/1.15	**0.32/0.59**	**0.66/1.11**	**0.42/0.81**

5.2.2. Time and space consumption

This paper compares the proposed model with six open-source approaches to evaluate the time and space efficiency. Table 7 details the total number of parameters and training speed (epoch 400 and batch size 64) for each model. The results show that the proposed model has the fewest parameters while providing the best prediction performance, except for the Social LSTM model. Although the training speed of this model is slower than that of other simpler models, the significant improvement in time and performance compared to the baseline STGAT model demonstrates the effectiveness of this approach in this paper.

Table 8 compares CUDA memory consumption during training and validation on an NVIDIA 2080TI GPU. The results indicate that the proposed model demonstrates strong space efficiency, achieving superior performance with lower memory usage than other models. This comparison of time and space consumption further underscores the effectiveness of the proposed model.

Table 6

Ablation test results (the best results are in bold)

Approaches	ADE/FDE					
	ZARA2	ZARA1	UNIV	HOTEL	ETH	AVG
STGAT-20V-20	0.30/0.61	0.34/0.68	0.56/1.20	0.33/0.62	0.71/1.26	0.45/0.87
GRU-LSTM-MLP2	0.30/0.62	0.33/0.68	0.55/1.18	0.32/0.62	0.69/1.19	0.44/0.86
GRU-LSTM-MLP1	0.30/0.61	0.34/0.68	0.55/1.17	0.33/0.64	0.70/1.22	0.44/0.86
GRU-LSTM	0.30/0.61	0.35/0.70	0.54/1.17	0.32/0.62	**0.65**/1.23	0.43/0.86
OURS-20V-20	**0.28/0.57**	**0.32/0.65**	**0.53/1.15**	**0.32/0.59**	0.66/**1.11**	**0.42/0.81**

Table 7

Number of parameters for 6 different approaches (parameter in piece, speed in second).

Approaches	Parameters	Speed
Social-LSTM [23]	264069	476.731
Social-Attention [34]	874949	9892.505
CIDNN [35]	1138228	1678.859
Social-GAN-20V-20 [29]	3,907330	128.567
STGAT-20V-20 [7]	44630	3380.0126
OURS-20V-20	20374	12175.480

Table 8

The comparison of CUDA memory usage (in MB)

Approaches	Training	Evaluation
Social-LSTM [23]	1252.01	922.75
Social-Attention [34]	2415.93	1734.35
CIDNN [35]	2208.31	2208.31
Social-GAN-20V-20 [29]	6213.89	1499.47
STGAT-20V-20 [7]	7134.54	1598.04
OURS-20V-20	2134.91	1153.44

Pedestrians in crowded scenes may have complex movement patterns, including forming groups, following other pedestrians, and changing directions to avoid collisions. Thus, in this test, visualizing the pedestrian trajectory predictions of the model (OURS-20V-20) and STGAT (STGAT-20V-20) in different scenarios.

Figure 4 shows some results on the scene with two or three pedestrians. The model can be seen that the model generates shorter prediction lines than the STGAT model. The reason is that the model first uses GRU for rapid and short-term prediction and then utilizes LSTM for prediction. However, the trajectories predicted by this paper are closer to the real situation, especially when two pedestrians walk side by side or move in the opposite direction of the crowd. The model captures more interactive information and can produce socially acceptable trajectories.

Figure 5 presents the results in complex scenarios. A comparison between Fig. 5a and Fig. 5d reveals that the approach effectively avoids collisions between pedestrians and stationary vehicles, producing socially acceptable outcomes. In more complex scenarios, comparisons between Fig. 5b and Fig. 5e, as well as Fig. 5c and Fig. 5f, demonstrate that the prediction accuracy of the proposed model surpasses that of the STGAT model. The proposed approach accurately simulates pedestrian interactions in complex environments, successfully avoiding collisions with static objects. These results illustrate that the proposed approach produces more accurate and socially acceptable predictions. Additionally, in the last row of Fig. 5., even in incorrect predictions where human-human and human-vehicle collisions occur, the predictor in this paper can still determine the future movement trends.

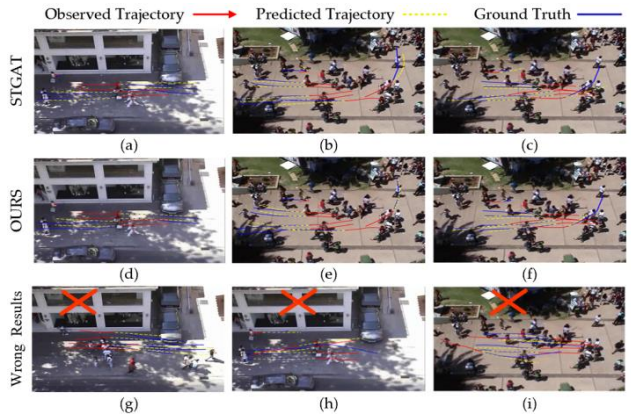

Fig. 4 – Some visual results of approach in this paper and STGAT on simple scene (with two or three pedestrians).

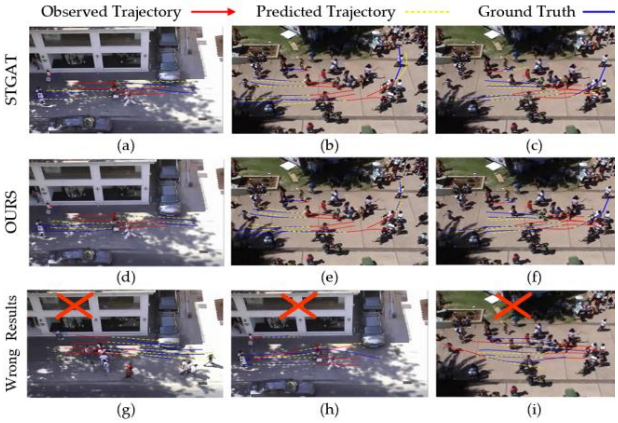

Fig. 5 – Some visual results of approach in this paper and STGAT on complex scene (with more pedestrians).

6. DISCUSSION AND CONCLUSIONS

This paper integrates MLPs, GRUs, and LSTM to enhance the original STGAT model for pedestrian trajectory prediction. The improved model incorporates an MLP into the GAT to reduce feature redundancy and replaces each LSTM with a GRU-LSTM to increase prediction accuracy. The model has been evaluated on the ETH and UCY datasets, demonstrating superior performance compared to 10 state-of-the-art approaches regarding average ADE and average FED. Visual results further confirm the model's ability to extract pedestrian motion information and accurately predict trajectories in simple and complex scenarios. Although some complex scenes lead to incorrect predictions, the approach can still accurately forecast the general movement trends of pedestrians.

Considering that the approach sometimes predicts wrong trajectories in the case of complex scenes, the future work will focus on extracting the complex interaction of multiple pedestrians.

ACKNOWLEDGEMENTS

This work is supported by the National Key R & D Program of China (No.2021YFF0603904), the National Natural Science Foundation of China (No.61771155), and the Fundamental Research Funds for the Central Universities (No.3072022TS0601).

REFERENCES

[1] Hopfield JJ. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences. 1982; 79(8): 2554–2558.
[2] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation. 1997; 9(8): 1735–1780.
[3] Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation [preprint], arXiv:1406.1078; 2014.
[4] Scarselli F, Gori M, Tsoi AC. The graph neural network model. IEEE Transactions on Neural Networks. 2008; 20(1): 61–80.
[5] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks [preprint], arXiv:1609.02907; 2016.
[6] Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks [preprint], arXiv:1710.10903; 2017.
[7] Huang Y, Bi H, Li Z, Mao T, Wang Z. STGAT: Modeling spatial-temporal interactions for human trajectory prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019, pp. 6272–6281.
[8] Wu H, Chen Z, Sun W, Zheng B, Wang W. Modeling trajectories with recurrent neural networks. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI). 2017, pp. 3083–3090.
[9] Shibata K, Goto K. Emergence of flexible prediction-based discrete decision making and continuous motion generation through actor-Q-learning. In: 2013 IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL). 2013.
[10] Al-Molegi A, Jabreel M, Ghaleb B. STF-RNN: Space time features-based recurrent neural network for predicting people next location. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI). 2016.

[11] Tang Y, Wang Y, Liu C, Yuan X, Wang K, Yang C. Semi-supervised LSTM with historical feature fusion attention for temporal sequence dynamic modeling in industrial processes. Engineering Applications of Artificial Intelligence. 2023; 117: 105547.
[12] Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. California Univ San Diego La Jolla Inst for Cognitive Science; 1985, pp. 318–362.
[13] Alahi A, Goel K, Ramanathan V, Robicquet A, Fei-Fei L, Savarese S. Social LSTM: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, pp. 961–971.
[14] Pfeiffer M, Schwesinger U, Sommer H, Galceran E, Siegwart R. Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2016, pp. 2096–2101.
[15] Bartoli F, Lisanti G, Ballan L, Del Bimbo A. Context-aware trajectory prediction. In: 2018 24th International Conference on Pattern Recognition (ICPR). 2018, pp. 1941–1946.
[16] Xu K, Qin Z, Wang G, Huang K, Ye S, Zhang H. Collision-free LSTM for human trajectory prediction. In: MultiMedia Modeling: 24th International Conference, MMM 2018. Bangkok, Thailand; February 5–7, 2018. Proceedings, Part I 24, pp. 106–116.
[17] Choi I, Song H, Yoo J. Deep learning based pedestrian trajectory prediction considering location relationship between pedestrians. In: 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC). 2019, pp. 449–451.
[18] Liu Y, Wang Z, Zheng B. Application of regularized GRU-LSTM model in stock price prediction. In: 2019 IEEE 5th International Conference on Computer and Communications (ICCC). 2019, pp. 1886–1890.
[19] Islam MS, Hossain E. Foreign exchange currency rate prediction using a GRU-LSTM hybrid network. Soft Computing Letters. 2021; 3: 100009.
[20] Kianimoqadam A, Lapp J. Calculating the view factor of randomly dispersed multi-sized particles using hybrid GRU-LSTM recurrent neural networks regression. International Journal of Heat and Mass Transfer. 2023; 202: 123756.
[21] Sari Y, Arifin YF, Novitasari N, Faisal MR. Deep learning approach using the GRU-LSTM hybrid model for air temperature prediction on daily basis. International Journal of Intelligent Systems and Applications in Engineering. 2022; 10(3): 430–436.
[22] Dan X. Spatial-temporal block and LSTM network for pedestrian trajectories prediction [preprint], arXiv:2009.10468; 2020.
[23] Yu C, Ma X, Ren J, Zhao H, Yi S. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In: Computer Vision – ECCV 2020: 16th European Conference. Glasgow, UK; August 23–28, 2020. Proceedings, Part XII 16, pp. 507–523.
[24] Kosaraju V, Sadeghian A, Martín-Martín R, Reid I, Rezatofighi H, Savarese S. Social-BiGAT: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In: Advances in Neural Information Processing Systems 32 (NeurIPS 2019).
[25] Zhu JY, Park T, Isola P, Efros A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. 2017, pp. 2223–2232.
[26] Widrow B, Hoff ME. Adaptive switching circuits. In: IRE WESCON Convention Record. Los Angeles, California; 1960, pp. 96–104.
[27] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. In: Proceedings of the fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings. 2011, pp. 315–323.
[28] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. 2015, pp. 448–456.
[29] Gupta A, Johnson J, Fei-Fei L, Savarese S, Alahi A. Social GAN: Socially acceptable trajectories with generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, pp. 2255–2264.
[30] Kingma DP, Ba J. Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR). 2015, pp. 1–15.
[31] Lerner A, Chrysanthou Y, Lischinski D. Crowds by example. Computer Graphics Forum. 2009; 26(3): 655–664.
[32] Pellegrini S, Ess A, Schindler K, Van Gool L. You'll never walk alone: Modeling social behavior for multi-target tracking. In: 2009 IEEE 12th International Conference on Computer Vision. 2009, pp. 261–268.
[33] Stone M. Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society: Series B (Methodological). 1974; 36(2): 111–147.
[34] Vemula A, Muelling K. Social attention: Modeling attention in human crowds. In: 2018 IEEE international Conference on Robotics and Automation (ICRA). 2018, pp. 4601–4607.
[35] Xu Y, Piao Z, Gao S. Encoding crowd interaction with deep neural network for pedestrian trajectory prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, pp. 5275–5284.
[36] Pang SM, Cao JX, Jian MY, Lai J, Yan ZY. BR-GAN: a pedestrian trajectory prediction model combined with behavior recognition. IEEE Transactions on Intelligent Transportation Systems. 2022; 23(12): 24609–24620.
[37] Sun Y, He T, Hu J, Huang H, Chen B. Socially-aware graph convolutional network for human trajectory prediction. In: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). 2019, pp. 325–333.
[38] Chen Y, Liu C, Shi B, Liu M. CoMoGCN: Coherent motion aware trajectory prediction with graph representation [preprint], arXiv:2005.00754; 2020.
[39] Sun J, Jiang Q, Lu C. Recursive social behavior graph for trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, pp. 660–669.

Received March 27, 2024