A Spatio-temporal Graph Network Allowing Incomplete Trajectory Input for Pedestrian Trajectory Prediction

Juncen Long, Gianluca Bardaro, Simone Mentasti, and Matteo Matteucci

***Abstract*— Pedestrian trajectory prediction is important in the research of mobile robot navigation in environments with pedestrians. Most pedestrian trajectory prediction algorithms require as input complete historical trajectories. If a pedestrian is unobservable in any frame in the past, then its historical trajectory becomes incomplete and the algorithm does not predict its future trajectory. To address this limitation, we propose STGN-IT, a spatio-temporal graph network allowing incomplete trajectory input. STGN-IT is able to predict the future trajectories of pedestrians with incomplete historical trajectories. STGN-IT uses the spatio-temporal graph with an additional encoding method to represent the historical trajectories and observation states of pedestrians. Moreover, STGN-IT introduces static obstacles in the environment that may affect the future trajectories as nodes to further improve the prediction accuracy. A clustering algorithm is also applied in the construction of spatio-temporal graphs. Experiments on public datasets show that STGN-IT outperforms state-of-the-art algorithms. Code will be released upon publication.**

I. INTRODUCTION

Many pedestrian trajectory prediction algorithms tried to help robots navigate in human-robot coexistence environments. However, almost all existing algorithms do not predict the future trajectories of pedestrians with incomplete historical trajectories. Specifically, when a pedestrian is unobservable in any frame in the past, its historical trajectory is defined as an incomplete trajectory, and almost existing algorithms do not predict the future trajectory of this pedestrian. This is not a serious problem for existing algorithms, as most of them use datasets labeled in the top-down view for training and evaluation. In particular, as shown in Fig. 1, pedestrians are not easily obscured in top-down view, so trajectories are almost always complete.

Many mobile robots collect information from egocentric view sensors rather than top-down view sensors, and pedestrians are more likely to be obscured in the egocentric view, as shown in Fig. 1. Therefore, the proportion of incomplete trajectories in the egocentric view dataset is higher than in the top-down view dataset. As a result, the shortcoming of existing algorithms that do not predict incomplete historical trajectories can occur frequently during robot navigation.

As shown in Fig. 2a, the historical trajectory of pedestrian 1 is complete and the historical trajectories of pedestrians 2 and 3 are incomplete, while a robot may collide with pedestrian 2. As shown in Fig. 2b, the prediction mode of almost all existing algorithms, the filtration mode [1], focuses on pedestrians with complete historical trajectories,

The authors are with the Department of Electronics, Information and Bioengineering, Politecnico di Milano, Milan, 20133, Italy. {juncen.long; gianluca.bardaro; simone.mentasti; matteo.matteucci}@polimi.it.

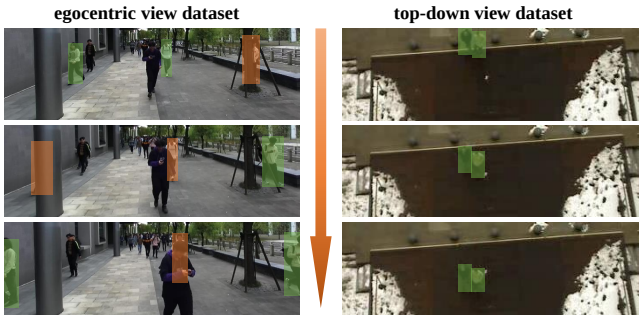

Fig. 1. Comparison of an egocentric view dataset (STCrowd) and a top-down view dataset (ETH). Green boxes indicate pedestrians are observable and orange boxes indicate pedestrians are obscured. Pedestrians are more likely to be obscured in the egocentric view than in the top-down view.

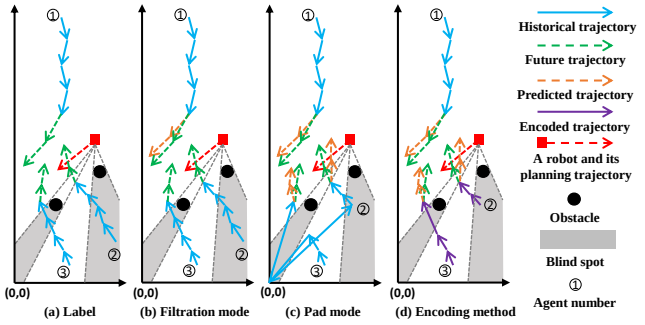

Fig. 2. Label and Prediction results for filtration mode, pad mode, and encoding mode with incomplete trajectories. Incomplete trajectories are not predicted in filtration mode. Incomplete trajectories are predicted in pad mode with unobservable positions set to 0. The encoding method encodes the observation state of positions.

so it only predicts the future trajectory of pedestrian 1. As shown in Fig. 2c, the prediction mode we used in this paper, the pad mode, is able to consider pedestrians with incomplete historical trajectories. The pad mode predicts the future trajectories of both pedestrians and represents the pedestrian's position as (0,0) when it is obscured. In this case, the pad mode is safer than the filtration mode as its prediction can indicate a possible collision for the robot. Therefore, it is ideal to train networks and evaluate the performance of algorithms in pad mode.

The incomplete historical trajectory in pad mode can be misinterpreted by algorithms that the pedestrian has moved from its original position to the position (0,0). Through the ablation study conducted in this paper, we find that this misinterpretation can reduce the performance of algorithms.

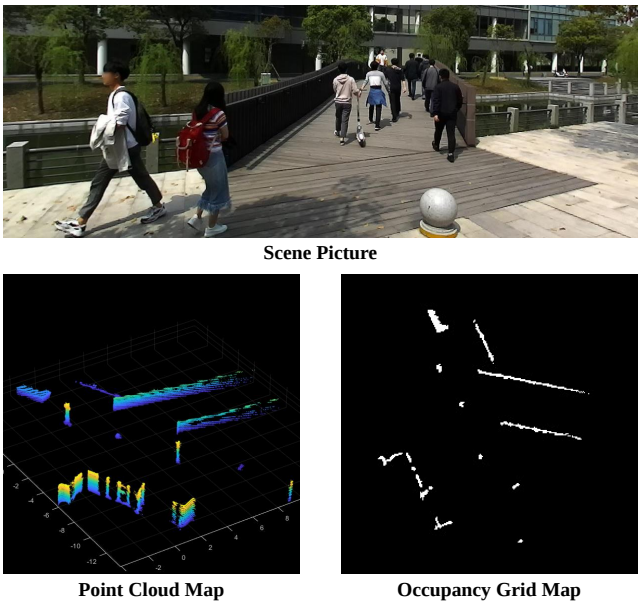

Fig. 3. The scene picture (top), point cloud map (bottom left), and occupancy grid map (bottom right) of a scene in the STCrowd dataset. The occupancy grid map can be automatically generated from the point cloud map.

To address the limitations of existing algorithms in dealing with incomplete historical trajectories, we designed a spatio-temporal graph network allowing incomplete trajectory input (STGN-IT) to predict the future trajectories of pedestrians. As shown in Fig. 3, STGN-IT obtains static obstacle information from the occupancy grid map, which can be automatically generated by the point cloud data. Thus, STGN-IT is more flexible than algorithms using semantic maps, which need to be manually labeled. In addition, STGN-IT uses spatio-temporal graphs to represent pedestrian and obstacle information and uses an encoding method to cope with the incomplete parts of historical trajectories. We train and evaluate STGN-IT with state-of-the-art algorithms on the public dataset STCrowd (STC) [2].

The main contributions of this paper are as follows.

1) Focusing on the incomplete trajectories that often occur in the egocentric view, we use the pad mode to deal with this situation and propose an encoding method to better handle incomplete trajectories with pad mode. With our parameter settings, the algorithm using pad mode is able to predict the trajectory of a pedestrian 1.2 seconds after observing it, rather than the 3.2 seconds required by most existing algorithms. As the case of pedestrian 2 shown in Fig. 2, the shorter response time makes our method more suitable for mobile robot navigation than the existing algorithms.
2) We designed STGN-IT, a spatio-temporal graph network allowing incomplete trajectory input. The density-based spatial clustering of applications with noise (DBSCAN) algorithm [3] is used to adjust the order of nodes in the spatio-temporal graph to help the network extract interaction features. The occupancy grid map is used to describe the environment to help the algorithm predict reasonable trajectories.

The paper is structured as follows; in Section 2, we present an overview of pedestrian trajectory prediction algorithms and datasets, focusing on available datasets and algorithms used for this task. In Section 3.A, we formulate the problem. From Section 3.B, we present the STGN-IT algorithm with its pipeline in detail. In Section 4, we compare quantitatively and qualitatively the performance of different algorithms in different prediction modes with the ablation study.

II. RELATED WORKS

Datasets for pedestrian trajectory prediction have different perspectives. For top-down view datasets, ETH [4] and UCY [5] are used most frequently. In recent years, SDD [6] has become popular, which has more data and complex environments. For egocentric view datasets, KITTI [7] and BDD100K [8] use sensors carried on vehicles, while SiT [9] and STC [2] use sensors carried by small robots or pedestrians. In the STC dataset, the sensors are carried on a static bracket.

There are many existing algorithms based on long short-term memory (LSTM) networks [10] for trajectory prediction [11]. Given that the gated recurrent unit (GRU) [12] has fewer parameters and similar performance to LSTM, some algorithms utilize GRU to predict trajectories [13]. Some algorithms also use encoder-decoder structures to improve the performance [14]. Social-VRNN [15] and Social-BiGAT [16] use the encoder–decoder structure to encode social interaction between pedestrians. A distribution discrimination method based on the encoder–decoder structure, DisDis [17], was proposed to learn the behavior pattern of each person.

Considering that interactions between pedestrians may occur at the end of the trajectories, some algorithms started to use the bidirectional LSTM (Bi-LSTM) and bidirectional GRU (Bi-GRU) to extract trajectory features [18], [19]. To better model interactions between pedestrians, some algorithms construct matrices using information of other pedestrians around a pedestrian, and extract features from matrices by neural networks. Social-LSTM [1] is the most representative one, where the velocities and positions of the surrounding pedestrians are embedded within a 3D matrix by square segmentation. Except for that, SS-LSTM [20] constructs the matrix using ring segmentation, and FSP [21] constructs the matrix using relative positions.

Some algorithms use spatio-temporal graphs to represent historical trajectories of pedestrians [22]. Typically, each node in the spatio-temporal graph represents a pedestrian, and the edges represent the correlations between pedestrians [23]. Social-STGCNN [24], a popular spatio-temporal graph based algorithm, uses relative velocity to represent edges. Then, features in the spatio-temporal graphs can be extracted by neural networks based on different structures [25]. For example, STAGP [26] uses convolutional neural networks to extract features, GST [27] uses recurrent neural networks, and STAR [28] uses transformers. All of the algorithms mentioned above use filtration mode in prediction and do

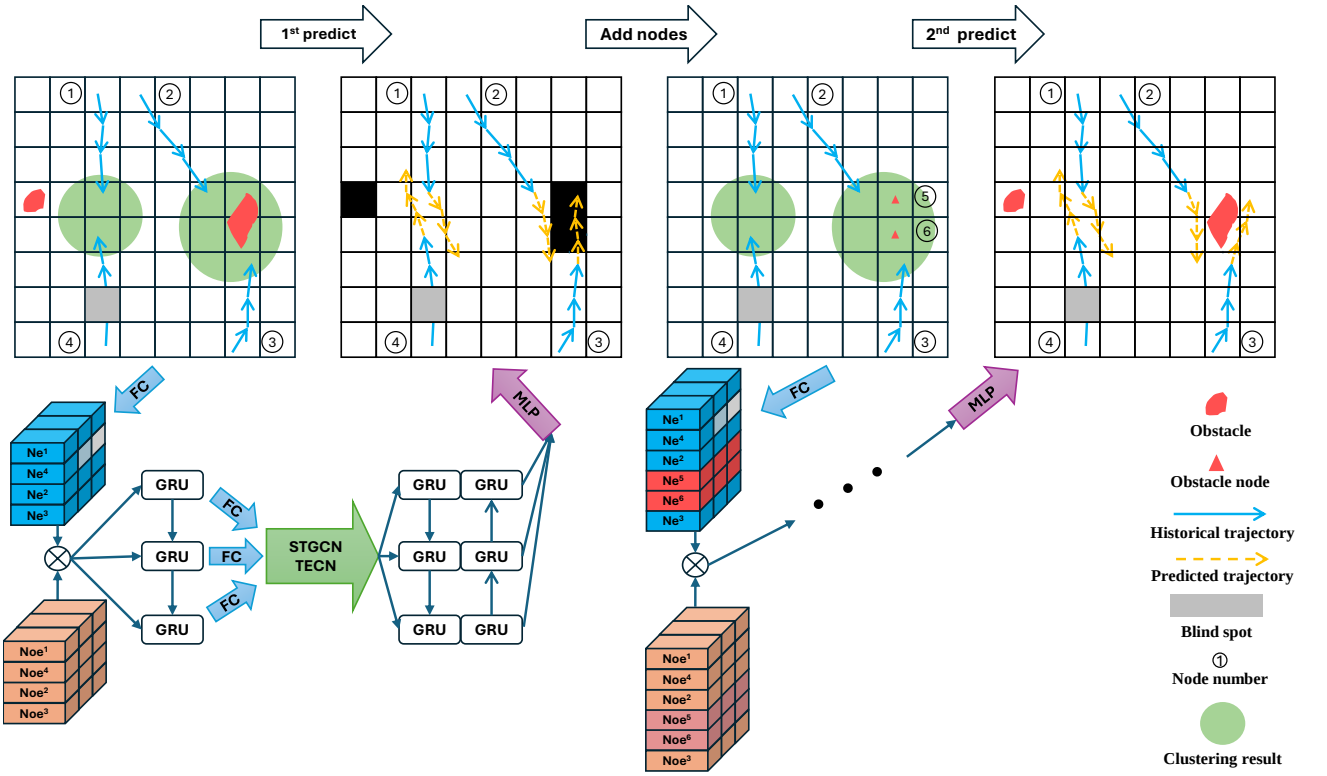

Fig. 4. The STGN-IT algorithm includes two predictions. The first prediction is used to search for obstacles that may affect pedestrians in the environment. After adding these obstacles to the spatio-temporal graph, the second prediction is performed to output the final trajectory prediction. In each prediction, the positions and observation states of the pedestrians are encoded by fully connected layers and fed into GRU networks. The spatio-temporal graph convolution network (STGCN), the time-extrapolator convolution network (TECN) and Bi-GRU networks are used to extract interaction features of pedestrians.

not allow input of incomplete historical trajectories, while STGN-IT allows it.

III. METHODOLOGY

A. Problem Formulation

The position information of pedestrian i at time t are represented as $X_t^i = [x_t^i, y_t^i]$. Further, $X_t^i = [x, y]$ when the pedestrian is observable with position $[x, y]$, and $X_t^i = [\text{Nan}, \text{Nan}]$ when the pedestrian is not observable. Suppose there are m pedestrians in the scene at time t_0, then their historical position information $H_{t_0}^{1:m}$ and ground-truth future position information $F_{t_0}^{1:m}$ can be represented as follows:

$$H_{t_0}^i = \left\{X_{t_0}^i, X_{t_0-1}^i, \ldots, X_{t_0-T_{obs}+1}^i\right\} \tag{1}$$

$$H_{t_0}^{1:m} = \left\{H_{t_0}^1, H_{t_0}^2, \ldots, H_{t_0}^m\right\}. \tag{2}$$

$$F_{t_0}^i = \left\{X_{t_0+1}^i, X_{t_0+2}^i, \ldots, X_{t_0+T_{pred}}^i\right\} \tag{3}$$

$$F_{t_0}^{1:m} = \left\{F_{t_0}^1, F_{t_0}^2, \ldots, F_{t_0}^m\right\}. \tag{4}$$

where T_{obs} and T_{pred} are the time step number of the historical and predicted trajectories, respectively. The algorithm outputs future trajectory predictions $\hat{F}^{1:m}$ for m pedestrians based on their historical trajectory information and environmental information. The loss function of STGN-IT tries to minimize the average displacement error (ADE) between the prediction $\hat{F}^{1:m}$ and the ground-truth trajectory $F_{t_0}^{1:m}$.

B. Algorithm Structure

The structure and prediction process of STGN-IT are shown in Fig. 4. The STGN-IT algorithm makes two predictions for a scene. In the first prediction, the network directly predicts the future trajectories of pedestrians without using environmental information. Since the first predicted trajectory does not consider environmental information, obstacles near it can affect the actual future trajectory of the pedestrian, and these obstacles need to be noticed. Therefore, obstacles near the first predicted trajectory are added to the spatio-temporal graph as nodes, and the spatio-temporal graph with obstacle information is used as input for the second prediction. Due to the use of environmental information, the trajectory of the second prediction is more accurate, and it is also the final output of STGN-IT.

The architecture of STGN-IT contains four modules. During the prediction, the spatio-temporal graph construction module generates the spatio-temporal graph and the corresponding matrices with the DBSCAN clustering algorithm, then the observation state encoding module encodes the matrices based on the pedestrian observation states, and finally the trajectory prediction module predicts the trajectories

TABLE I
CODE RULES FOR No_t^i

ON_t^i	ON_{t-1}^i	No_t^i
True	True	[1,1,1,1]
True	False	[1,1,0,0]
False	/	[0,0,0,0]

TABLE II
CODE RULES FOR Eo_t^i

ON_t^i & ON_t^j	ON_{t-1}^i & ON_{t-1}^j	Eo_t^{ij}
True	True	[1,1,1,1]
True	False	[1,1,0,0]
False	/	[0,0,0,0]

based on the encoded features. Between two predictions, the obstacle addition module adds obstacles near the predicted trajectories to the spatio-temporal graph as nodes.

C. Spatio-temporal Graph Construction Module

When constructing the spatio-temporal graph, we consider both position and velocity information. In the spatio-temporal graph, node N_t^i represents the information of a pedestrian or an obstacle at time t, and edge E_t^{ij} represents the correlation between N_t^i and N_t^j at time t. Suppose $\Delta X_t^i = X_t^i - X_{t-1}^i$ is the velocity of N_t^i at time t, then N_t^i and E_t^{ij} can be represented as follows:

$$N_t^i = [X_t^i, \Delta X_t^i] \tag{5}$$

$$E_t^{ij} = [X_t^i - X_t^j, \Delta X_t^i - \Delta X_t^j] \tag{6}$$

Then, we use the DBSCAN clustering algorithm to adjust the order of the nodes in the matrix, making the interactions between the nodes easier to detect. Specifically, the DBSCAN algorithm clusters nodes based on the density of nodes in different regions, and nodes that are classified into the same class are adjacent to each other in the matrix.

As shown in Fig. 4, the node order in the matrix is $(N^1, N^4, N^2, N^5, N^6, N^3)$ instead of $(N^1, N^2, N^3, N^4, N^5, N^6)$, because N^1 and N^4 are classified in a class, while N^2, N^5, N^6 and N^3 are classified in another class. The obstacles N^5 and N^6 are close to the pedestrians N^3 and N^2, which may have an impact on their future trajectories, so making these four nodes neighboring in the matrix is beneficial for networks to extract the features.

D. Observation State Encoding Module

When node N_t^i is not observable, we let $X_t^i = [0, 0]$ and $\Delta X_t^i = \Delta X_{t+1}^i = [0, 0]$. To allow the network to distinguish whether a node is not observable or truly in position (0,0), we design an encoding rule to describe the observation state. Specifically, we add two vectors, No_t^i and Eo_t^{ij}, to describe the availability of the data. Define ON_t^i as the observation state of N_t^i, the specific rules are shown in Table I and Table II.

Then, for the nodes, we use the fully connected layers to combine the information from N_t^i and No_t^i to obtain the feature Nf_t^i:

$$Ne_t^i = \phi^{ne}(N_t^i; W_{ne}) \tag{7}$$

$$Noe_t^i = \phi^{noe}(No_t^i; W_{noe}) \tag{8}$$

$$Nf_t^i = Ne_t^i \odot Noe_t^i \tag{9}$$

where ϕ^{ne} and ϕ^{noe} are the fully connected layers with an input dimension of 4 and an output dimension of n_{en}, and $\odot$ represents the Hadamard product.

By using other two fully connected layers ϕ^{ee} and ϕ^{eoe}, and a similar process to (7)-(9), we can also obtain the edge feature Ef_t^{ij}. Then, by embedding them into the corresponding places of the matrices, the new node matrix Vf and the adjacency matrix Af can be constructed. The dimension of Vf is $[T_{obs}, m, n_{en}]$ and the dimension of Af is $[T_{obs}, m, m, n_{en}]$.

E. Trajectory Prediction Module

In order to reduce the influence of missing positions on the trajectory feature extraction, GRU networks are first used to compensate for the missing position information by utilizing the features from previous frames. The compensation node matrix Vfc can be constructed as follows:

$$Hvfc = \text{GRU}^{vf}(Hvf, Vf; W_{vf}) \tag{10}$$

$$Vfc = \phi^{vfc}(Hvfc; W_{vfc}) \tag{11}$$

where GRU^{vf} is a GRU network with an input layer dimension of n_{en} and hidden state dimension of n_{gru}, and ϕ^{vfc} is the fully connected layer with an input dimension of n_{gru} and an output dimension of n_{en}. Hvf is the initial hidden state of GRU^{vf}, and $Hvfc$ is the stack of hidden states at each step of GRU^{vf}. Similarly, we construct the compensation adjacency matrix Afc by another network GRU^{af} and ϕ^{afc}, and a similar process to (10)-(11). Vfc has the same dimension as Vf, and Afc has the same dimension as Af.

Then, we use the spatio-temporal graph convolution network (STGCN) and the time-extrapolator convolution network (TECN) to extract features from the Vfc and Afc matrices. Briefly, based on 2D convolutional networks with residual structures, STGCN extracts the interaction features between nodes at each time step from Vfc and Afc, then TECN extracts the temporal correlation features of each node from $Vstg$. STGCN and TECN have the same structures as [24] but different parameters. The process is shown as follows:

$$Vstg = \text{STGCN}(Vfc, Afc; W_{stgcn}) \tag{12}$$

$$Vp = \text{TECN}(Vstg; W_{tecn}) \tag{13}$$

where STGCN is an STGCN model with a kernel size of n_{stg} and TECN is a three-layer TECN model with a kernel size of n_{te}. The node prediction matrix Vp has a dimension of $[T_{pred}, m, n_{de}]$.

Finally, a Bi-GRU network is utilized to extract the features in Vp to obtain the matrix GVp, and then a multi-layer perception (MLP) network is utilized to decode GVp and output the final prediction. The process is shown as follows:

$$Sp_{1:T_{pred}} = \text{BiGRU}^{gvp}(Hgvp, GVp; W_{pv}) \tag{14}$$

$$\Delta\hat{X}_t = \text{MLP}^{sp}(Sp_t; W_{sp}) \quad (t = 1, 2, \ldots, T_{pred}) \tag{15}$$

$$\hat{X}_t = \hat{X}_{t-1} + \Delta\hat{X}_t \quad (t = 1, 2, \ldots, T_{pred}) \tag{16}$$

where BiGRU^{gvp} is a Bi-GRU network with an input layer dimension of n_{de} and hidden state dimension of n_{gru}, and MLP^{sp} is a three-layer MLP network with an input dimension of $2 * n_{gru}$ and an output dimension of 2. $Hgvp$ is the initial hidden state of BiGRU^{gvp}, and $Sp_{1:T_{pred}}$ is the stack of hidden states at each step of BiGRU^{gvp}. $\Delta\hat{X}_t$ and $\hat{X}_t$ are the displacement and position of pedestrians predicted by the network at time t, respectively.

F. Obstacle Addition Module

As shown in Fig. 4, after the first prediction, the obstacle addition module adds obstacle nodes to the spatio-temporal graph based on the occupancy grid map and the predicted trajectories. Suppose the set of predicted trajectories in the first prediction is $\hat{\boldsymbol{X}}$. The obstacle positions in the occupancy grid map can be represented as (x^i_{obs}, y^i_{obs}). The set of obstacles added to the spatio-temporal graph $\boldsymbol{Obs}$ is defined as follows:

$$\boldsymbol{Obs} = \left\{(x^i_{obs}, y^i_{obs}) | f_{\text{mindis}}((x^i_{obs}, y^i_{obs}), \hat{\boldsymbol{X}}) < od\right\} \tag{17}$$

where the function $f_{\text{mindis}}(p, S)$ is defined as the minimum distance from the point p to all points in the set S, and od is the minimum distance to add an obstacle as a node.

Since the obstacles in $\boldsymbol{Obs}$ are close to the predicted trajectories, they affect the future trajectories of pedestrians. After adding them to the spatio-temporal graph as nodes, STGN-IT makes the second prediction, which considers the influence of the static obstacles and has a higher accuracy.

IV. EXPERIMENTS AND ANALYSIS

A. Evaluation Metrics

We use average displacement error (ADE) and final displacement error (FDE) to evaluate the performance of algorithms, which are defined as follows:

$$\text{ADE} = \frac{\sum_{i=1}^{m}\sum_{t=1}^{T_{\text{pred}}}\left\|\hat{X}^i_t - X^i_t\right\|_2}{m * T_{\text{pred}}} \tag{18}$$

$$\text{FDE} = \frac{\sum_{i=1}^{m}\left\|\hat{X}^i_{T_{\text{pred}}} - X^i_{T_{\text{pred}}}\right\|_2}{m} \tag{19}$$

where $\|\cdot\|_2$ denotes the Euclidean norm. ADE and FDE are measured in meters.

B. Dataset and Parameter Settings

The STC dataset contains over 5000 annotated trajectories in more than 10 scenes. We use the raw LIDAR data to create point cloud maps and further generate occupancy grid maps. We use trajectories in the STC dataset to train and evaluate the performance of STGN-IT and state-of-the-art algorithms.

The STC dataset annotates the data at a rate of 2.5 FPS. We set the observation time as 3.2 seconds, the prediction time as 4.8 seconds, $T_{obs} = 8$ and $T_{pred} = 12$, which is the same setting as state-of-the-art algorithms. As defined in (20) and (21), the conditions for STGN-IT to perform trajectory prediction for pedestrian i are that it is observable in the latest frame and is observable for over 2 of the past 8 frames. With this rule, STGN-IT predicts the trajectory of a pedestrian 1.2 seconds after observing it, rather than the 3.2 seconds required by most existing algorithms. The shorter response time makes STGN-IT more suitable for mobile robot navigation than other existing algorithms.

$$X^i_0 \neq [\text{Nan}, \text{Nan}] \ \& \sum_{t=-T_{obs}+1}^{0} f_{\text{exist}}(X^i_t) > 2 \tag{20}$$

$$f_{\text{exist}}(X) = \begin{cases} 1 & \text{if } X \neq [\text{Nan}, \text{Nan}] \\ 0 & \text{if } X = [\text{Nan}, \text{Nan}] \end{cases} \tag{21}$$

The structural parameters of the network are set as $n_{en} = 9$, $n_{de} = 7$, $n_{gru} = 64$, $n_{stg} = 7$, $n_{te} = 3$. The distance parameters are set as $od = 0.8$. The learning rate is set to 0.001, the batch size is set to 16, and the number of epochs is set to 200.

To further evaluate the influence of missing trajectories on the algorithm performance, we randomly removed about 10% of the samples from the original STC dataset and generated a new dataset, STC-c dataset. We only remove the inputs of the samples and make sure that the inputs still satisfy (20) and (21) after removing the data, so that the STC dataset and the STC-c dataset still have the same labels.

C. Experiment Settings

We compare STGN-IT with the following state-of-the-art algorithms: STIGCN [22] (2024), IMGCN [29] (2024), MSRL [30] (2023), Social-Implicit [31] (2022), SGCN [32] (2021), Social-STGCNN [24] (2020).

Referring to [33], for the state-of-the-art algorithms, we randomly sample 3 times and select the samples with the best metrics, that is, minADE_3 and minFDE_3. For STGN-IT, we output three candidate trajectories based on the possible directions of pedestrians.

Even though calculating ADE and FDE with filtration mode are the metrics used in most publications, as demonstrated in Fig. 1 and Fig. 2, pad mode is safer for robot

TABLE III

THE minADE_3/minFDE_3 RESULTS ON THE STC DATASET. **BOLD** AND UNDERLINED MARK THE BEST AND SECOND-BEST RESULTS.

	STC,f-f ↓	STC,p-p ↓	STC-c,p-p ↓
Social-STGCNN	0.68/1.18	0.74/1.32	0.84/1.49
SGCN	0.38/0.69	0.43/0.77	0.67/1.19
Social-Implicit	0.38/0.76	0.55/1.06	0.80/1.50
MSRL	0.41/0.77	0.58/0.92	0.69/0.98
IMGCN	0.42/0.76	0.49/0.89	0.52/0.93
STIGCN	0.39/0.71	0.46/0.83	0.51/0.89
STGN-IT	**0.30/0.56**	**0.35/0.62**	**0.35/0.64**

navigation rather than filtration mode. To better evaluate the performance of the algorithms, we train and evaluate them with two modes. Only Pedestrians with complete historical trajectories are predicted in filtration mode, and pedestrians that satisfy (20) are predicted in pad mode.

In the following, we refer to the condition "p-p" as the algorithm trained and tested with pad mode, and refer to the condition "f-f" as the algorithm trained and tested with filtration mode.

D. Quantitative Experiments and Analysis

Table III shows the ADE and FDE of algorithms evaluated in three prediction conditions. The performance of all algorithms decreases from "STC,f-f" to "STC,p-p" to "STCc,p-p", due to the increasing incomplete parts of the trajectories between these three conditions. However, the performance degradation rate differs greatly among different algorithms. Social-Implici has almost twice the metrics in "STCc,p-p" as 'STC,f-f", and even metrics for algorithms that are less affected, such as STIGCN and IMGCN, also increase by over 22%. This situation leads to the second-best algorithm being different in three prediction conditions. STGN-IT, on the other hand, has a performance degradation rate of about 15%, which is the smallest among all the algorithms. Also, STGN-IT has the best ADE and FDE in all three conditions, meaning that STGN-IT maintains the best performance regardless of the completeness of the trajectory.

E. Ablation Study

We explore the influence of different modules on the performance of STGN-IT through an ablation experiment with the following algorithms:

1) STGN-IT without adding obstacle nodes from occupancy grid maps. (w/o map)
2) STGN-IT without observation state encoding. (w/o code)
3) STGN-IT without clustering process. (w/o cluster)

Table IV shows the ADE and FDE of algorithms evaluated in different prediction conditions. The least affected algorithm is STGN-IT w/o code in condition "f-f", which makes sense because in condition "f-f" all input trajectories are complete and the observation state encoding is redundant. In addition to this case, the deletion of any module significantly reduces the performance of the algorithms, resulting in an increase of ADE and FDE by at least 20%.

TABLE IV

THE minADE_3/minFDE_3 RESULTS OF ABLATION STUDY. **BOLD** AND UNDERLINED MARK THE BEST AND SECOND-BEST RESULTS.

	STC,f-f ↓	STC,p-p ↓
STGN-IT w/o map	0.41/0.78	0.43/0.79
STGN-IT w/o code	0.32/0.58	0.46/0.85
STGN-IT w/o cluster	0.40/0.74	0.48/0.93
STGN-IT	**0.30/0.56**	**0.35/0.62**

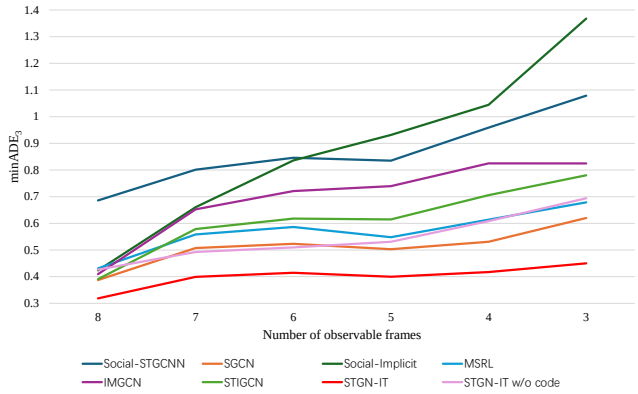

Fig. 5. The minADE_3 results of algorithms in 'STC,p-p" condition with different numbers of observable frames. Due to encoding the observation state, the performance decay of STGN-IT is less than other algorithms when the number of observable frames decreases.

Fig. 5 shows the ADE results of STGN-IT, STGN-IT w/o code, and the state-of-the-art algorithms in 'STC,p-p" condition with different numbers of observable frames. The number of samples with observable frames from 3 to 8 in 'STC,p-p" condition is $[1702, 2060, 2659, 3601, 5522, 43177]$. When the number of observable frames decreases, STGN-IT has less performance degradation than the other algorithms, while STGN-IT w/o code has more performance degradation than STGN-IT, which proves the efficiency of observation state encoding.

We also compare the performance of STGN-IT w/o map on the ETH and UCY dataset. These datasets do not provide point cloud data, so we use STGN-IT w/o map rather than STGN-IT, as STGN-IT has to use occupancy grid maps created by point cloud data.

Table V shows the ADE and FDE of the algorithms

TABLE V

THE minADE_3/minFDE_3 RESULTS ON THE ETH AND UCY DATASET IN CONDITION "F-F". **BOLD** AND UNDERLINED MARK THE BEST AND SECOND-BEST RESULTS.

	eth ↓	hotel ↓	zara1 ↓	zara2 ↓
Social-STGCNN	0.88/1.55	0.60/1.10	0.48/0.87	0.43/0.77
SGCN	**0.72/1.39**	0.44/0.85	**0.39**/0.78	**0.29/0.59**
Social-Implicit	0.85/1.89	0.47/0.89	0.41/0.83	0.37/0.78
MSRL	0.85/1.82	0.64/1.43	0.42/0.87	0.90/1.96
IMGCN	0.85/1.40	0.46/0.83	0.41/0.81	0.34/0.66
STIGCN	0.78/1.49	0.44/0.83	0.40/**0.77**	0.50/0.97
STGN-IT w/o map	0.77/1.53	**0.42/0.75**	0.46/0.98	0.34/0.74

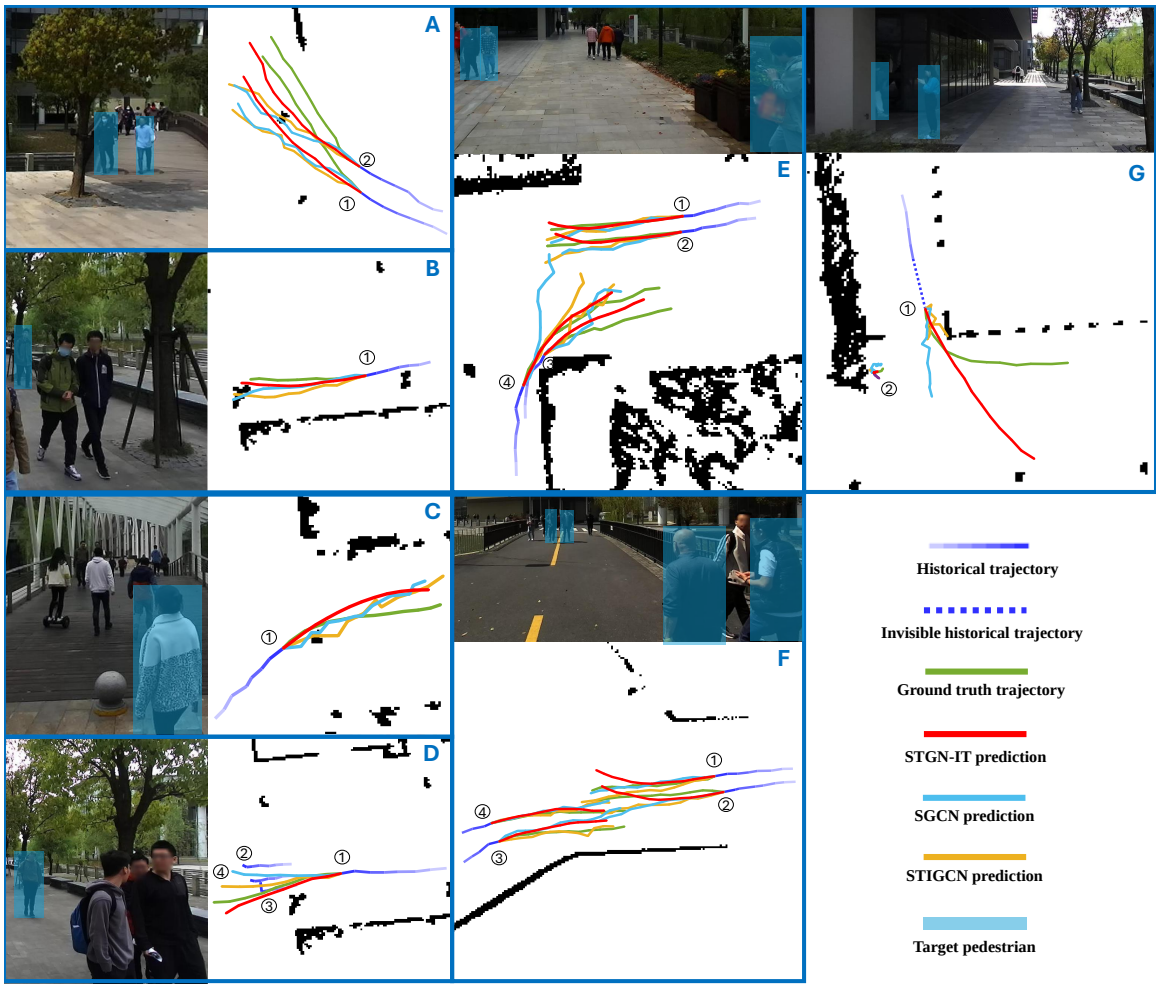

Fig. 6. The predictions of STGN-IT and some state-of-the-art algorithms. The trajectories predicted by the STGN-IT are more reasonable, as the trajectories avoid static obstacles in scenes A, B, C and avoid other pedestrians in scenes D, E, F. With incomplete trajectory input in scene G, predictions of STGN-IT are smoother and more reasonable.

evaluated in the ETH and UCY datasets. Due to the lack of occupancy grid maps and the use of condition "f-f", the performance of STGN-IT w/o map is not superior compared to SGCN, IMGCN and STIGCN in some datasets. This result is consistent with the results shown in Tables III and IV, where STGN-IT w/o map performs worse than these algorithms in "STC, f-f" condition.

F. Qualitative Analysis

The prediction results of some scenes are shown in Fig. 6. In scenes A, B, and C, the trajectories predicted by some state-of-the-art algorithms cross with static obstacles, while the predictions of STGN-IT do not. This is because STGN-IT uses the occupancy grid map as the input, reducing the probability of predicting collision trajectories.

STGN-IT also successfully predicts the interactions between pedestrians. In scene D, pedestrians 2, 3, and 4 stop on the road, and pedestrian 1 bypasses them, and STGN-IT successfully predicts the bypass trajectory. In scene E, four pedestrians meet at an intersection, and the STGN-IT successfully predicts their turns, while the trajectories predicted by the SGCN collide. The same situation occurs in scene F, where only the trajectories predicted by STGN-IT avoid collisions, and the trajectories predicted by SGCN and STIGCN collide.

In scene G, when the trajectory of pedestrian 1 is partially missing due to column occlusion, the trajectories predicted by STGN-IT are smooth and roughly correct, while the trajectories predicted by STIGCN and SGCN are very unstable. The trajectory predictions for Pedestrian 2 demonstrate that all algorithms have good predictions for stationary pedestrians.

The qualitative analysis demonstrates that STGN-IT has good trajectory prediction performance and can predict trajectories that are smooth and close to the ground-truth labels.

V. CONCLUSION

In this paper, we propose a spatio-temporal graph network allowing incomplete trajectory input (STGN-IT) for pedestrian trajectory prediction. We focus on pad mode instead of filtration mode for pedestrian trajectory prediction because pedestrians are easily obscured and cause incomplete histor-

ical trajectories in the egocentric view of the mobile robot. In this context, we propose an encoding method to help STGN-IT better solve this problem. Our algorithm employs the DBSCAN clustering algorithm to adjust the order of nodes to help the network extract features. Moreover, occupancy grid maps of the environment are used to improve the reasonability of the trajectory predictions. Evaluations on the STC dataset show that the ADE and FDE of STGN-IT are at least 10% lower than the state-of-the-art algorithms.

ACKNOWLEDGMENT

Juncen Long acknowledges the support of the China Scholarship Council.

REFERENCES

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and 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.

[2] P. Cong, X. Zhu, F. Qiao, Y. Ren, X. Peng, Y. Hou, L. Xu, R. Yang, D. Manocha, and Y. Ma, "Stcrowd: A multimodal dataset for pedestrian perception in crowded scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 608–19 617.

[3] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[4] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 261–268.

[5] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in *Computer graphics forum*, vol. 26, no. 3. Wiley Online Library, 2007, pp. 655–664.

[6] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*. Springer, 2016, pp. 549–565.

[7] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[8] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636–2645.

[9] J. W. Bae, J. Kim, J. Yun, C. Kang, J. Choi, C. Kim, J. Lee, J. Choi, and J. W. Choi, "Sit dataset: socially interactive pedestrian trajectory dataset for social navigation robots," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[10] A. Graves and A. Graves, "Long short-term memory," *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

[11] R. Quan, L. Zhu, Y. Wu, and Y. Yang, "Holistic lstm for pedestrian trajectory prediction," *IEEE transactions on image processing*, vol. 30, pp. 3229–3239, 2021.

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[13] M. Archana, R. Viji, and S. Ganapathy, "A cnn-gru based hybrid approach for pedestrian trajectory prediction," in *2024 10th International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2024, pp. 1611–1616.

[14] Y. Xu, Z. Piao, and S. Gao, "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.

[15] B. F. de Brito, H. Zhu, W. Pan, and J. Alonso-Mora, "Social-vrnn: One-shot multi-modal trajectory prediction for interacting pedestrians," in *Conference on Robot Learning*. PMLR, 2021, pp. 862–872.

[16] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. Reid, H. Rezatofighi, and S. Savarese, "Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks," *Advances in neural information processing systems*, vol. 32, 2019.

[17] G. Chen, J. Li, N. Zhou, L. Ren, and J. Lu, "Personalized trajectory prediction via distribution discrimination," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 580–15 589.

[18] J. Sun, Q. Jiang, and 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.

[19] Y. Yao, E. Atkins, M. Johnson-Roberson, R. Vasudevan, and X. Du, "Bitrap: Bi-directional pedestrian trajectory prediction with multi-modal goal estimation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1463–1470, 2021.

[20] H. Xue, D. Q. Huynh, and M. Reynolds, "Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction," in *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2018, pp. 1186–1194.

[21] J. Long, J. Mei, and G. Ma, "Egocentric two-frame pedestrian trajectory prediction algorithm based on a panoramic camera," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–13, 2022.

[22] W. Chen, H. Sang, J. Wang, and Z. Zhao, "Stigcn: spatial–temporal interaction-aware graph convolution network for pedestrian trajectory prediction," *The Journal of Supercomputing*, vol. 80, no. 8, pp. 10 695–10 719, 2024.

[23] S. Haddad, M. Wu, H. Wei, and S. K. Lam, "Situation-aware pedestrian trajectory prediction with spatio-temporal attention model," *arXiv preprint arXiv:1902.05437*, 2019.

[24] A. Mohamed, K. Qian, M. Elhoseiny, and 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. 14 424–14 432.

[25] J. Lian, W. Ren, L. Li, Y. Zhou, and B. Zhou, "Ptp-stgcn: pedestrian trajectory prediction based on a spatio-temporal graph convolutional neural network," *Applied Intelligence*, vol. 53, no. 3, pp. 2862–2878, 2023.

[26] Z. Liu, L. He, L. Yuan, K. Lv, R. Zhong, and Y. Chen, "Stagp: Spatio-temporal adaptive graph pooling network for pedestrian trajectory prediction," *IEEE Robotics and Automation Letters*, 2023.

[27] Z. Huang, R. Li, K. Shin, and K. Driggs-Campbell, "Learning sparse interaction graphs of partially detected pedestrians for trajectory prediction," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1198–1205, 2021.

[28] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "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*. Springer, 2020, pp. 507–523.

[29] W. Chen, H. Sang, J. Wang, and Z. Zhao, "Imgcn: interpretable masked graph convolution network for pedestrian trajectory prediction," *Transportmetrica B: Transport Dynamics*, vol. 12, no. 1, p. 2389896, 2024.

[30] Y. Wu, L. Wang, S. Zhou, J. Duan, G. Hua, and W. Tang, "Multi-stream representation learning for pedestrian trajectory prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 3, 2023, pp. 2875–2882.

[31] A. Mohamed, D. Zhu, W. Vu, M. Elhoseiny, and C. Claudel, "Social-implicit: Rethinking trajectory prediction evaluation and the effectiveness of implicit maximum likelihood estimation," in *European Conference on Computer Vision*. Springer, 2022, pp. 463–479.

[32] L. Shi, L. Wang, C. Long, S. Zhou, M. Zhou, Z. Niu, and G. Hua, "Sgcn: Sparse graph convolution network for pedestrian trajectory prediction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8994–9003.

[33] P. Kothari, S. Kreiss, and A. Alahi, "Human trajectory forecasting in crowds: A deep learning perspective," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7386–7400, 2021.