Self-Growing Spatial Graph Networks for Pedestrian Trajectory Prediction

Sirin Haddad
Nanyang Technological University
siri0005@e.ntu.edu.sg

Siew-Kei Lam
Nanyang Technological University
asskLam@ntu.edu.sg

Abstract

Intelligent vehicles and social robots need to navigate in crowded environments while avoiding collisions with pedestrians. To achieve this, pedestrian trajectory prediction is essential. However, predicting pedestrians' trajectory in crowded environments is nontrivial as human-to-human interactions among the crowd participants influence their motion. In this work, we propose a novel end-to-end graph-centric gated learning model to estimate the existence of interactions between individuals. Accordingly, the model predicts pedestrians' future locations and velocities. Recent methods based on LSTM networks used thresholding techniques to define neighborhood boundaries and relationships. Other graph-structured methods grow edges in polynomial size. In contrast, our graph-based GRU network model employs an online data-driven criterion that can learn from interactions and grow connections between pedestrian nodes. The proposed model yields outperforming prediction accuracy over state-of-the-art works in two public datasets, i.e. *Crowds and SDD.*

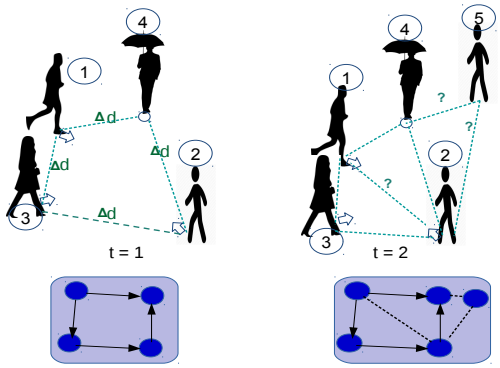


Figure 1. Relationship prediction between two pedestrians. The relationship in this context refers to the dynamical influence imposed by pedestrians onto each other. By visually tracking the relative distance Δd at time-steps t_1 and t_2, one can assess whether there is a potential influence between two pedestrians as indicated by the dotted lines tagged with $<?>$ between pair of pedestrians. Resolving relational assessment is depicted by establishing edges between the nodes, where each node refers to a pedestrian in the scene.

1. Introduction

Pedestrian trajectory prediction is essential for enabling intelligent vehicles and social robots to avoid collisions during navigation in crowded environments [24, 8, 1, 27]. However, pedestrian trajectory prediction in urban environments is a challenging task as human navigation decisions are influenced by their social interactions with other traffic participants. Besides, urban environments are characterized by a temporally varying number of pedestrians and motion dynamics, which further increases the difficulty in developing accurate prediction models.

From a cognitive perspective, crowds motion imitates a collective tendency towards self-organization and coordination of pedestrians motion inside the crowd [11]. We take our inspiration from this normative theory, to build a self-growing graph that estimates the potential for an interaction between two pedestrians and reflects that accordingly onto the spatial connectivity inside each graph step. Figure 1 shows a typical scene of pedestrians, each of them conducting different motion dynamics such as: running, walking, and standing. The proposed method aims to discover the relationship between the pedestrians, which is a manifestation of the social behavior among pedestrians (*i.e.* to avoid collisions) that impacts their subsequent trajectories.

Existing works attempt to learn the interactive context of urban environments using fixed spatial neighborhoods to provide local features of pedestrians social interactions [1]. The idea of defining neighborhood boundaries using the neural model was highlighted earlier in [29]. Nevertheless, few structured architectures developed this task on graphs considering the spatial relations between pedestrians [9, 32] using fixed parameters to establish neighborhoods. In this work, we propose a novel end-to-end graph-centric gated learning model to estimate the existence of interactions between individuals. Our interaction modeling is based on a principled understanding of social relations that governs potential interactions between pedestrians and inspired by

relational reasoning [25].

Previous approaches were conducted in offline learning setting such that the neighborhood size should be tuned to fit new scenes. For autonomous mobility in unknown environments, the prediction model will benefit from the ability to continually learn from incoming data streams. Learning from data streams is known as online learning [15, 2, 14], which in our context, reasons about future trajectory instantly based only on the current step, and withdrawing the previous steps from the model. However, applying this setting directly to Long-Short Term Memory (LSTM) provokes instability in its predictions, as the internal cell memory requires longer sequences to learn motion correlations that are embedded along consecutive time-steps.

In our case, maintain a continual long-term trajectory prediction. We ensure that pedestrians nodes are stateful by carrying their state through gated memory to mitigate the prediction instability problem and, at the same time, allow for an adaptive structure by enabling the spatial graph to grow dynamically while keeping the network trainable and majorly stable.

To the best of our knowledge, our work is the first to simultaneously estimate social relations between pedestrians while learning their dynamics using a dynamic incomplete graph structure.

Contribution Our work improves state-of-the-art pedestrian trajectory prediction in the following aspects:

1. In previous works, socially-focused methods employed an exclusive motion feature, i.e. velocity or location displacement, for prediction. In contrast, we predict multiple motion features at one pass to improve the trajectory forecast. Motion features can be better exploited, by combining future velocity predictions with future location prediction, rather than solely relying on the past location sequence.

2. We develop a self-learned criterion for growing graph topology under a variable number of nodes, reducing the message passing load between the nodes in the graph network.

3. We introduce spatio-temporal graphs in online continual learning by preserving the statefulness of nodes states with gated memory units. The graph creates its spatial and temporal structure with minimal data preprocessing per frame.

2. Related Works

Neighborhood selection for pedestrian trajectory prediction A plethora of research works approached pedestrian trajectory prediction from several perspectives including their social interaction modeling as means of understanding how pedestrians move with respect to each others motion [3, 7, 22, 28, 31, 24]. The social interaction centered approaches proved to be more applicable than focusing on pedestrian as an individual entity in the environment. The relational bias suggests that pedestrians tend to interact within small distances forming local neighborhoods. This assumption proved to be useful for crowd motion representation under limited scenarios. Recent advances in graph-structured learning conduct parameter sharing and parameters tying mechanisms on tabularized representation, to have a compact form of the variable size environments [13, 19]. However, these approaches resorted to a fixed grid-like neighborhood layout for passing messages between nodes, either by growing spatially in quadratic space [27] or locally set neighborhoods [1]. Other supervised models [20, 30] expand neighborhood concept from local scope to global scope such that parameter sharing between neighborhoods is achieved using gated memory cells.

Link Prediction in Graphs for neighborhood selection Link Prediction is a big task in data mining problems, such as recommender systems, social networks, and protein formation networks [5, 21, 17]. Nevertheless, the idea of predicting future links (or relations) in graphs has significant potential that covers graph-structured data in any domain. Studying crowd motion modeling as a graph completion problem finds a new area for link prediction. However, predicting the future association between pedestrians is non-trivial, as there have not been theoretical guarantees for an optimal solution. Fixed neighborhood-based heuristics have been empirically set to gain the best predictions, but as crowds motion is a dynamic system, pedestrians often have dynamic neighborhoods, which makes the Euclidean discretization a scene-specific solution. It is best to reason about pedestrian interaction based on graph structure and extend the neighborhood concept to the non-Euclidean zone.

In graph streams, it is more challenging to estimate association heuristic without having an initially connected graph to anticipate the missing links. We resort to learning adaptive-sized neighborhoods and estimate the association between the nodes, without specifying heuristics apriori. Existing works in pedestrian trajectory prediction are always based on ground-truth settings regarding the latent graph structure. Unlike the existing works, our graph model does not rely on any assumption, *e.g.* minimum Euclidean distance, to fixate the spatial connections. Instead, we permit a variable size and an arbitrary shape for the spatial neighborhoods from one time-frame to another to allow generalization in the social interaction modeling.

3. Proposed Method

3.1. Problem formulation

We formulate our pedestrian trajectory problem as follows: Let X be pedestrians trajectories, such that: $X = x_1, x_2, ..., x_n$, with n pedestrians. $\widetilde{X}$ are future trajectories, x_t^i is i-th pedestrian trajectory from time-step $t = 1$ until $t = t + obs$, given that obs is observation length. We observe 8 steps of each pedestrian trajectory and predict for the next 12 steps. Each predicted step is added to the first predicted point, $x_{t+pred}^i = x_{t+1}^i + x_{t+2}^i + ... + x_{t+l}^i$ to maintain consistency and dependency between predicted steps. Along with trajectory prediction, each pedestrian has its velocity prediction for each time-step and added to the respective predicted steps.

3.2. Method

In this section, we present our proposed models: M_{TV} and M_{TVP}, and M_{SGTV}, where SG stands for Self-Growing, T for trajectories, V for Velocity and P for Pooling layer. Thereby, any model containing T indicates its reliance on positional features and the same naming convention applies to the rest of the symbols.

Overall, we formulate the problem of growing graph as a graph-focused learning task: Assume Graph G comprised of nodes set N, temporal edges set Σ_T, and an unknown spatial edges set Σ_S, our task is to grow the spatial set to minimize the Euclidean error in nodes output predictions. Our learning objective for this task is formulated as follows:

$$min_G\{_{\Sigma_T,\Sigma_S}\}||\widetilde{X}_n - X_n||_2^2 \quad , \tag{1}$$

According to Graph Network (GN) [4], the prediction pipeline comprises two sets of functions: aggregators ρ and updaters ϕ. Aggregator functions are responsible for augmenting nodes and edges states before processing and updater functions are applied at nodes and edges to output their final states.

3.3. Centralized models

M_{TV} and M_{TVP} both process data on a set of disconnected nodes, withdrawing relationships from the graph structure. This design choice is a preliminary element to test the capability of the most basic recurrent structure. They train the data within a central Multi-Stacked GRU so that at each frame all the nodes trajectories and the output predictions are set in a tabulated form to ease indexing and retrieval of each node.

The minibatch models increase the observation length. As shown in Figure 2, the minibatch training takes 8 steps to predict the next 12 frames. They aggregate all nodes trajectories and passes them into the stacked GRU cell to get positional predictions:

$$\widetilde{X}_{\{t,t+l\}}, h_{\{t,t+l\}} = GRU(x_{\{t-l,t\}}, h_{\{t-l,t\}}), \tag{2}$$

As previous models calculate velocities of observed trajectories, we dedicate another GRU cell for predicting walking velocity V^*, preserving the existing hidden space and pass it again into the second GRU:

$$V = F(v_{n\times l}) \tag{3}$$

$$F(V) = W_{ve} * v + b_{ve} \tag{4}$$

$$V^*_{\{t,t+l\}}, h_{v\{t,t+l\}} = GRU(V, h_{\{_{t-l},t\}}), \tag{5}$$

Then the velocity predictions are summed with the respective positional predictions to update nodes future trajectories:

$$X^*_{\{t,t+l\}} = \widetilde{X}_{\{t,t+l\}} + V^*_{\{t,t+l\}} \quad , \tag{6}$$

Predicting new velocity at each step enables the model to continuously adapt to sudden changes in position data. It is beneficial to add the higher-order motion to positional prediction, as this combination pace the model performance for different walking velocities. Relying on position alone makes the model prone to significant errors, which can affect motion pattern understanding without adding extra cues and information.

As the multi-stacked GRU has 4 layers, it generates 4 mappings of every predicted sequence. Therefore, we used average pooling layer P to average the mappings before transforming to the output layer:

$$X^*_x = \frac{\sum_{i=1}^{4} X^*_{i\{t,t+l\}}}{4}, \tag{7}$$

Eventually, the network preserves statefulness of GRU cells over time, by adding the previous hidden states into the recent ones:

$$h_t = \sum_{i=1}^{t-1} h_i \quad , \tag{8}$$

3.4. Self-Growing Gated model

Self-Growing Graph Network aims to overcome the inefficiency issues due to reliance on stable, dense connections in graph-structured networks. We developed an efficient adaptive neighborhood approach, which can execute at runtime for an evolving graph. Learning in graph network uses edge features and nodes features to learn the global graph. The aggregation function accumulates respective temporal edge and node features in one message chunk and gets passed to the global stacked GRU cell, where features are set in layered order and information is propagated from one layer up to the next layer. Propagating messages in a stacked style at one global cell plays a core role in forming a hierarchical concept of the hidden representation.

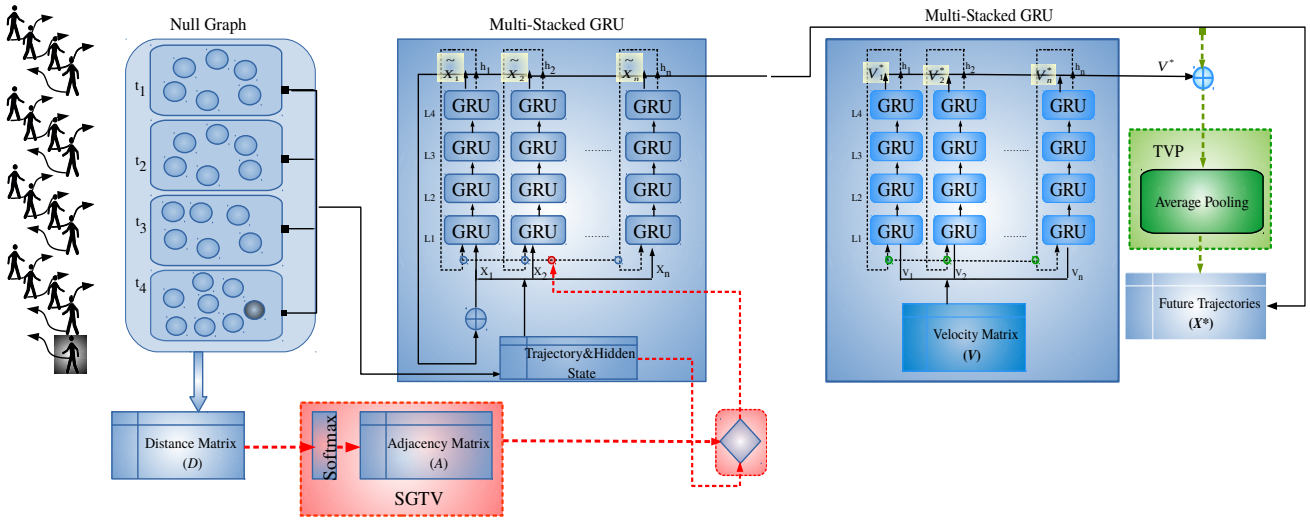


Figure 2. Structure and learning pipeline combining parts of the three models: M_{TV},M_{TVP} and M_{SGTV}. Each model has special components which are enclosed within dashed boxes and tagged with model name. The basic model M_{TV} operates without the average pooling layer and adjacency matrix, while M_{SGTV} works only without the pooling. The solid circles indicate pedestrian nodes, the diamond operator $\diamond$ illustrates the conditional selection of inputs to form a neighborhood Z with directed edges and the circled plus operator $\bigoplus$ is for addition of entries. X is the input trajectories set, V is the velocity set, $\widetilde{X}$ is the output of GRU, h_i is the input for initial hidden states of i-th trajectory, $h_i^{'}$ is the output hidden state of i-th trajectory. V^* is the predicted velocity and finally, X^* is the final predicted trajectory after the sum of V^* and $\widetilde{X}$. (Best viewed in color).

For our self-growing mechanism, we map graph G to the Adjacency matrix $A_{n\times n}$, in which the dimensions can be changed according to the maximum observed pedestrians. This is based on estimating adjacency potential for each of pair of pedestrians. Firstly, we initialize matrix to zeros as all nodes are disconnected, then the spatial edges are evaluated by distance matrix D, which stores distances between all nodes in the scene:

$$D^t_{n\times n} = \begin{cases} 0; & i = j \\ ||x^t_i - x^t_j||^2_2 & otherwise \end{cases} \tag{9}$$

Eq. (10) stores normalized Euclidean distances $\widetilde{D}^t_{n\times n}$ as follows:

$$\widetilde{D}^t_{n\times n} = \begin{cases} 0; & i = j, D^t_{(i,j)} = 0 \\ \frac{1}{D^t_{(i,j)}} & otherwise \end{cases} \tag{10}$$

The distance matrix gets embedded using linear function F, which uses weight matrix W_{de} and bias vector b_{de} to transform from distance space to embedded feature space:

$$F(D) = W_{de} * D + b_{de} \tag{11}$$

$$\hat{e} = F(\widetilde{D}^t_{n\times n}), \tag{12}$$

Similar to [12], the potential for adjacency is produced by passing $\widetilde{D}$ through $softmax$ layer which is suitable for transforming distances into probability space between $(0, 1)$ and one digit of precision. However, our adjacency matrix is not necessarily symmetric. Depending on the estimation of influence on a directed graph, the influence can be determined in one-way.

$$A^t_{n\times n} = softmax(\widetilde{D}^t_{n\times n}), \tag{13}$$

Typically, $A^t_{n\times n}$ gets its probabilistic entries rounded to the nearest integer, so that it is used for selecting the embedded spatial edges' features $\hat{e}$. This selection forms the neighborhood Z of which all common nodes are considered influential to node i:

$$Z_t = \left\{\hat{e}_{(i,j)}; \quad if \quad A^t_{(i,j)} = 1, \right. \tag{14}$$

After estimating the global neighborhood for every node, GRU cell receives nodes embeddings $X_{\{1,t\}}$ together with neighborhood feature embeddings, such that $P_{\{1,t\}}$ are all the observed trajectories:

$$x_{\{1,t\}} = F(P_{\{1,t\}}) \tag{15}$$

$$\widetilde{X}_{\{t+1,t+l\}}, h_{\{t+1,t+l\}} = GRU(x_{\{1,t\}}, Z_t, h_{\{1,t\}}), \tag{16}$$

The flow continues as described in Eq. (5) and Eq. (6) previously.

Dataset	ETH	hotel	Students	UCY	Zara
Avg velocity	2.5	1.2	0.9	1.6	1.4

Table 1. Average walking velocity over various subsets in Crowds dataset (m/s)

Dataset	Death Circle	Gates	Hyang
Avg velocity	1.0	1.2	1.0

Table 2. Average walking velocity over various subsets in Stanford drone dataset (m/s)

4. Experiments

In this section, we present experiments results for our models. Crowd motion can be highly dynamic, hence we conducted experiments to demonstrate our Graph Network accuracy, adaptability to variations in social interactions and efficiency in comparison with state-of-the-art models. All the experiments and testing in this study were carried out using a desktop computer with Intel Core-i5 3.1 GHz CPU and 16 GB memory running Ubuntu 16.04 operating system. All models were implemented in PyTorch.

4.1. Datasets

We evaluate our models on the set of videos included in the TrajNet challenge [23]. Subsets such as ETH and UCY videos are commonly tested across the literature, and they only illustrate pedestrian-wise interactions. We choose to include additional subsets to indicate how well our models perform under different dynamic characteristics and walking patterns. Tables 1 and 2 provide pedestrians average walking velocity in Crowds and SDD datasets respectively. We calculated the average velocities as L2 displacements along with trajectory steps over the frame rate. Velocity average can hint on the nature and intensity of social interactions which distract pedestrians and therefore make them walk slower.

1. Stanford Drone Dataset (SDD) [22] is a heterogeneous dataset, containing different categories including pedestrians, bikers, cyclists, and skateboarders. While pedestrians and bicyclists are prevalently present in every scene, the other categories exhibit faster motion, and this poses a stronger influence on walking pedestrians. We choose to include this dataset in our evaluation as it is more challenging for modeling the correlation between different dynamic patterns.

2. Crowds [18] is a widely tested dataset, containing 8 videos taken from 4 urban scenes, ETH, Hotel, Zara and UCY, making over 6K annotated frames and 1530 pedestrians that have different styles of walking and interactions with each other.

   Table 1, illustrates that the dataset embraces different ranges of walking velocity profiles ranging from [0.5 – 2.5] m/s on average. Unlike Table 2 which shows that for subsets like Gates, Death circle, and Hyang pedestrians share similar walking velocities around 1.0 m/s.

4.2. Training Setup

We run the network based on Mean Squared Error (MSE) loss function Eq. (17). MSE provides the advantage of squaring L2 errors, which greatly penalizes estimation errors and therefore leads to faster convergence in streamed and semi-streamed models, where frequent updates create variations in loss curve, such that it ends up converging to a high error value. In our models, mini-batch training struggles to produce a smooth and convex loss curve. However, we ensure that the model converges to the lowest local minimum point possible.

$$\mathcal{L} = \frac{\sum_{i=1}^{N} ||X_i^* - X_i||_2^2}{N}, \tag{17}$$

For hyper-parameters settings, we set 1e-05 for the learning rate, which is smaller than other baselines [1] by two magnitudes. We also set gradient clipping at 10 which is scheduled at every batch. Embedding size is 128 per node vector and 256 per edge vector. The dropout parameter is 1e-04, and the lambda regularization parameter is 5e-04. The batch size is 1, trajectory observation length is 8 steps and prediction length is 12 steps. We re-evaluated [8] model on our observation and prediction lengths, however, we referred to [27, 1] reported evaluation results due to lack of available implementation.

All the weight matrices and bias vectors used in the multi-stacked cell and linear layers are initialized according to a uniform distribution of range [-0.008, 0.008] which is scaled by feature embedding size of 128. The weight decay rate is 1e-4. The number of layers in a multi-stacked cell is 4 and so is the batch size.

We also reset the graph every 100 batches by deleting nodes and edges of pedestrians that no longer exist in the scene. This is to maintain the running time and training parameters size.

4.3. Evaluation Metrics

We calculate Euclidean L2 norms to measure displacements between ground-truth and predicted trajectories. Our evaluations are based on the following two metrics:

1. *Average Displacement Error (ADE)* which is the average $\mathcal{L}_2$ displacements along all predicted steps:

$$E = \frac{\sqrt{\sum_{i=1}^{N} \sum_{j=1}^{l} (\widetilde{X_i^j} - X_i^j)^2}}{N * l}, \tag{18}$$

2. *Final Displacement Error (FDE)* which is the $\mathcal{L}_2$ displacements at the final point:

$$E = \frac{\sqrt{\sum_{i=1}^{N}(\widetilde{X_i} - X_i)^2}}{N} \quad . \tag{19}$$

Other methods such as [1, 13] calculate root square for each predicted trajectory. In our case, we calculate the square root of all Euclidean errors once all the predictions are generated. We noticed that this calculation shows statistical stability. It is well-representative of average errors over generated predictions and does not get biased by the drastic, sudden variances in the Euclidean error range, due to the instability induced by the gated models in end-to-end sequence learning, for encoders like GRU or LSTM.

4.4. Baselines

1. *Structural-RNN* [13] is a spatio-temporal LSTM graph method. They illustrate their graph in several interactive contexts for activity forecasting, such as the interaction between human and static objects. However, they preserve the bipartiteness property for fixed relationships between the nodes. In crowd modeling, arbitrary graphs can be better at generalizing over the exchanged influence between pedestrians.

2. *S-LSTM* [1] dedicates LSTM for every pedestrian, and pool their states before predicting future steps. Their method only combines features of pedestrians who are found occupants of common neighborhood space. The neighborhood and occupancy grid sizes are set empirically for attaining the best results over ETH and UCY datasets.

3. *S-GAN* [8] also dedicates LSTM for every pedestrian, deployed within Encoder-Decoder architecture for generating future predictions using GAN network. Each pedestrian gets multiple sampled trajectories, and eventually, the least erroneous sample is selected for presentation. The authors publish their code so we retrained their model with a non-variational setting to compare with our work.

4. *Desire* [16] is an RNN-based variational model for predicting objects trajectories in heterogeneous interactions. They collect multimodal data streams, including static and dynamic features in a centralized gated encoder-decoder unit. Their model is conditioned over an unknown distribution for learning its parameters.

5. *GRE* Gated Relation Encoder [6], a recent work inspired by [25] generalized relational inference network. It is an LSTM-based method for predicting pedestrians social and contextual relationships. From the relational perspective, this method tracks the relationship importance by encoding correlations between visual motion features and static environment features for all possible pairings of objects and selects the relations with the highest potential.

6. *Social Attention S-Attn* is a graph-based full-connectivity method that assesses the influence of social interactions with weighted edges. However, they grow edges constantly between all pedestrians.

4.5. Quantitative Results

It is observed from the quantitative ADE results in Table 3 that outperforming results were achieved in M_{SGTV} over ETH, Hotel and UCY sets. At the second place comes, M_{TV} and M_{TVP}, which present comparable performance, yet they produce the highest prediction errors in Univ subset. They omit relationships modeling from the graph, realizing a use-case of the stacked central GRUs as explained earlier by Section 3.3. The average pooling layer in M_{TVP} model works on global social interactions. Unlike social pooling in [1, 26, 10], our social pooling layer combines nodes future representations as shown in Figure 2. This is a new order of pooling deployment in deep networks, as the social pooling of existing states, enables the model to implicitly understand neighborhood impact on a given pedestrian. Predictions pooling advances the model understanding of interaction future evolution, and take feedback from the predicted interactions.

At the bottom line, all of our models outperform the baseline models due to the usage of the stacked gated memory cells. The improvement is achieved due to the inclusion of walking velocity for every pedestrian as well as predicting future velocity to estimate relative future locations concerning the previous time-step and velocity. The self-growing mechanism provides an advantage above the basic model M_{TV}, and the fully-connected graph models [13], [27] in which the graph connections grow densely with time. Overloading the graph with node-wise connections may include unrelated nodes together and complicate the graph structure. Likewise, neglecting the social factors would limit the model from learning the collision avoidance pattern conducted naturally among pedestrians.

Compared to Crowds dataset, Stanford Dataset includes a more challenging setting, as it has objects with significantly variant dynamics. Each object conducts different patterns of dynamics. Hence, a prediction algorithm ought to adapt to these variations in one scene. For example, pedestrians are walking while cyclists are making displacements at a faster rate, and potentially both pedestrian and cyclist meet at the same scene zone. The model tasks here are understanding each object motion and the variation in rela-

tionships between two different objects to make plausible predictions.

We conducted a short ablation study to discover the impact of changing observation to prediction lengths ratio. We evaluated S-GAN and M_{SGTV} models observing 4 steps and predicting the next 8 steps as shown in Table 4. It manifests the effect of increasing the gap between observation and prediction lengths, which explains why both models performed worse than they did when observing 8 steps and predicting 12 steps as of Table 3.

Table 5 shows the best prediction results at 1/5th of image resolution. We pick DESIRE-SI-IT4 from [16] which iterates over the data sequence 4 times. Both [16] and [6] embeds visual scene semantics besides the social interactions. We acknowledge the impact of including scene features in the literature, yet our proposed models focus on encoding pedestrian dynamics and inferring their potential relationships. Both GRE [6] and M_{SGTV} are interaction-aware frameworks that infer relationships using gated models.

Our model dedicates two structured GRU cells for learning the correlation between pedestrians displacements and velocities, respectively. Since we sum predicted motion features and feed them again into GRU, this expands GRU spatio-temporal learning into the future besides its learning from past features which is useful for improving predictions iteratively. M_{SGTV} produced higher average errors than M_{TV}.

This is due to treating social interaction in isolation from environment static context. Besides, some pedestrians behavior is challenging as they are inconsiderate of the social norms, so their behavior counters the interactions happening in the context. This situation challenges the interaction-oriented modeling, *i.e.* M_{SGTV}, as pedestrians appear less interactive with each other.

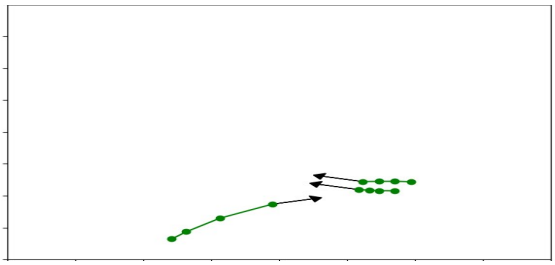
The best result for M_{TV} is found under the BookStore subset (3.45 pixels), while M_{SGTV} achieves the best results in the DeathCircle (3.55 pixels) scene and Gates (5.72 pixels).

4.6. Qualitative Analysis

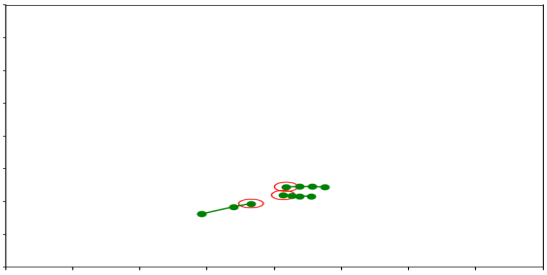
Figure 3 contains snapshots of Self-Growing mechanism performance when predicting pedestrians trajectories in Zara dataset.

Figure 3(a) illustrates a scene where the model slows down the trajectories as it anticipates a potential influence between pedestrians facing each other and directly reflects that on future pedestrian velocity. The motion direction of the predicted trajectory also manifests the understanding of collision-avoidance norm.

Existing literature addressed the LSTM tendency toward trajectory overestimation, where pedestrians are predicted with faster walking speed in recurrent models. Figure 3(b) shows velocity range estimation through circles when pedestrians are facing each other, such that the velocity extent is not exceeding the personal space of pedestrians. This is due to the benefit of understanding how people walk in relative to each other.



(a)



(b)

Figure 3. Illustration of M_{SGTV} estimation of interactions in Crowd dataset. The dotted spline shows ground-truth trajectory. Arrows indicate predicted trajectory direction in collision avoidance situation.

4.7. Running time

Pedestrian count can be used as a factor to examine the algorithm efficiency boundaries. In general, the more objects are observed, the longer a forward pass takes. We examined 3 of our proposed models over 20 runs for scenes with 20, 40, and above 60 pedestrians, respectively. Results are displayed in Table 6, and illustrate the running time per frame along with pedestrians counts in the graph. For 20 pedestrians, our models are nearly comparable with the SGAN model. Knowing that M_{TVP} pools at each time step and SGAN is set to generate a single trajectory sample, our models pace the performance downgrade. The running time increases by up to 0.5x, *i.e.* time taken in M_{SGTV} for 40 pedestrians versus 60 pedestrians and more. Whereas, in SGAN, the running time inclines faster as pedestrians count increases. To support this argument, we synthesized a crowd of 74 pedestrians and ran both M_{TVP} and SGAN,

Model	ETH	Hotel	Zara1	Zara2	Zara3	UNIV	AVG
Structural-RNN [13]	2.72/4.60	0.85/1.35	1.05/2.20	1.60/3.50	–	1.45/3.00	1.53/ 2.93
S-LSTM [1]	1.09/2.35	0.79/1.76	0.47/1.00	0.56/1.17	–	0.67/1.40	0.91/1.54
S-GAN [8]	0.81/1.52	0.72/1.61	0.34/0.69	0.42/0.84	–	0.60/1.26	0.58/1.18
S-Attn [27]	0.39/3.74	0.29/2.64	0.20/0.52	0.30/2.13	–	0.33/3.92	0.30/2.59
M_{TV}	**0.04/0.14**	0.04/0.15	0.05/**0.25**	0.08/0.31	0.08/**0.29**	0.04/0.20	0.06/0.22
M_{TVP}	0.04/0.15	0.04/0.15	0.06/0.26	0.08/0.31	0.08/0.31	0.04/0.20	0.06/0.23
M_{SGTV}	0.04/0.15	**0.04/0.15**	**0.05**/0.26	**0.07/0.30**	**0.08**/0.30	**0.04/0.15**	**0.05/0.21**

Table 3. Prediction errors (meters) over *Crowds dataset* Results Format: ADE/FDE. Each pedestrian is observed for 8 frames (3.2 seconds) and predicted for future 12 steps (4.8) seconds. The dash (–) replaces missing results for the baseline methods

Model	ETH	Hotel	Zara1	Zara2	Zara3	UNIV	AVG
S-GAN [8]	0.89/1.60	0.69/0.93	0.43/1.00	0.43/0.89	–	0.50/1.07	0.59/1.10
M_{SGTV}	0.05/0.22	0.05/0.23	0.10/0.30	0.11/0.35	0.12/0.38	0.16/0.46	0.10/0.32

Table 4. Empirical analysis of observation/prediction lengths gap. Prediction errors (meters) over *Crowds dataset* given that observation length is changed to 4 steps (1.6 seconds) and prediction length is 8 steps (3.2 seconds).

Model	Best	AVG
Desire-SI-IT4 [16]	7.55	–
S-LSTM	9.85	–
GRE-MC-5 [6]	5.99	–
M_{TV}	**1.22**	**1.78/4.72**
M_{SGTV}	1.26	6.43/14.85

Table 5. Comparing best results achieved in *DESIRE* model with our models at 1/5 resolution. M_{TV} and M_{SGTV} illustrates the lowest FDE for length of 3.2 seconds while [16] displays the best predictions on their model for length of 3 seconds.

Model	c=20	c=40	c>60
M_{TV}	0.2456	0.3619	0.5010
M_{TVP}	0.2590	0.3560	**0.4895**
M_{SGTV}	**0.2314**	**0.3520**	0.7568
SGAN	0.2965	0.5895	1.0525

Table 6. Running time in seconds for batch size = 1. c denotes pedestrians count.

where we could observe that running cost escalates at larger crowds. This situation occurred due to SGAN pooling the features of all the possible pairs of pedestrians, which enlarges the hidden space quickly.

Edge Growth Rate According to a fully-connected graph, each node has $(n-1)$ edges, where n is the number of nodes. The complete graph ends up with $n^2 - n$ edges. Theoretically, M_{SGTV} worst-case scenario might reach a polynomial of the second degree. Through experiment, our model grows edges linearly, due to the usage of softmax for evaluating a relationship existence based on the relative distances.

5. Conclusion

In this paper, we proposed an approach for growing spatial relationships in graph-structured networks for modeling pedestrian motion and interactions. This approach is considered an approximation of the ground-truth interactions, such that anticipated relations between pedestrians rely on a data-driven criterion that is stemmed from their velocities and relative distances along time. The presence of links in the graph is variant, resulting in spatially dynamic routes that carry the hidden representation between the nodes. Although our method generates relationships with different degrees of confidence, it does not further exploit this information for weighting the impact of different pedestrians. In future work, we plan to develop a weighting mechanism to evaluate the influence and accordingly improve the pedestrians associations decision.

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*, pages 961–971, 2016. 1, 2, 5, 6, 8

[2] A. Aroor, S. L. Epstein, and R. Korpan. Online learning for crowd-sensitive path planning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1702–1710. International Foundation for Autonomous Agents and Multiagent Systems, 2018. 2

[3] F. Bartoli, G. Lisanti, L. Ballan, and A. Del Bimbo. Context-aware trajectory prediction. *arXiv preprint arXiv:1705.02503*, 2017. 2

[4] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive

biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. 3

[5] H. Chen, X. Li, and Z. Huang. Link prediction approach to collaborative filtering. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '05)*, pages 141–142, June 2005. 2

[6] C. Choi and B. Dariush. Looking to relations for future trajectory forecast. *arXiv preprint arXiv:1905.08855*, 2019. 6, 7, 8

[7] T. Fernando, S. Denman, S. Sridharan, and C. Fookes. Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. *Neural networks*, 108:466–478, 2018. 2

[8] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. *arXiv preprint arXiv:1803.10892*, 2018. 1, 5, 6, 8

[9] S. Haddad, M. Wu, H. Wei, and S. K. Lam. Situation-aware pedestrian trajectory prediction with spatio-temporal attention model. *Computer Vision Winter Workshop (CVWW)*, pages 4 – 13, 2019. 1

[10] I. Hasan, F. Setti, T. Tsesmelis, A. Del Bue, F. Galasso, and 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*, pages 6067–6076, 2018. 6

[11] D. Helbing, L. Buzna, A. Johansson, and T. Werner. Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions. *Transportation science*, 39(1):1–24, 2005. 1

[12] I. Henrion, J. Brehmer, J. Bruna, K. Cho, K. Cranmer, G. Louppe, and G. Rochette. Neural message passing for jet physics. 2017. 4

[13] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, 2016. 2, 6, 8

[14] S. Kim, S. J. Guy, W. Liu, R. W. Lau, M. C. Lin, and D. Manocha. Predicting pedestrian trajectories using velocity-space reasoning. In *Algorithmic Foundations of Robotics X*, pages 609–623. Springer, 2013. 2

[15] J. F. Kooij, F. Flohr, E. A. Pool, and D. M. Gavrila. Context-based path prediction for targets with switching dynamics. *International Journal of Computer Vision*, 127(3):239–262, 2019. 2

[16] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. 2017. 6, 7, 8

[17] C. Lei and J. Ruan. A novel link prediction algorithm for reconstructing proteinprotein interaction networks by topological similarity. *Bioinformatics*, 29(3):355–364, 12 2012. 2

[18] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007. 5

[19] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan. Semantic object parsing with graph lstm. In *European Conference on Computer Vision*, pages 125–143. Springer, 2016. 2

[20] X. Liang, X. Shen, D. Xiang, J. Feng, L. Lin, and S. Yan. Semantic object parsing with local-global long short-term memory. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3185–3193, 2016. 2

[21] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007. 2

[22] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016. 2, 5

[23] A. Sadeghian, V. Kosaraju, A. Gupta, S. Savarese, and A. Alahi. Trajnet: Towards a benchmark for human trajectory prediction. *arXiv preprint*, 2018. 5

[24] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, and S. Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. *arXiv preprint arXiv:1806.01482*, 2018. 1, 2

[25] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976, 2017. 2, 6

[26] D. Varshneya and G. Srinivasaraghavan. Human trajectory prediction using spatially aware deep attention models. *arXiv preprint arXiv:1705.09436*, 2017. 6

[27] A. Vemula, K. Muelling, and J. Oh. Social attention: Modeling attention in human crowds. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7. IEEE, 2018. 1, 2, 5, 6, 8

[28] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. Who are you with and where are you going? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1345–1352. IEEE, 2011. 2

[29] F. Yang, H. Saikia, and C. Peters. Who are my neighbors?: A perception model for selecting neighbors of pedestrians in crowds. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, pages 269–274. ACM, 2018. 1

[30] Y. Yuan, X. Liang, X. Wang, D.-Y. Yeung, and A. Gupta. Temporal dynamic graph lstm for action-driven video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1801–1810, 2017. 2

[31] F. Zanlungo, T. Ikeda, and T. Kanda. Social force model with explicit collision prediction. *EPL (Europhysics Letters)*, 93(6):68005, 2011. 2

[32] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. *arXiv preprint arXiv:1903.02793*, 2019. 1